JOINT INTEGRATION TEST FACILITY (JITF) and

National Imagery and Mapping Agency (NIMA) Integration Test Facility (ITF)

Department of Defense Intelligence Information System (DoDIIS)

INTEGRATION REQUIREMENTS and EVALUATION PROCEDURES

Version 4.2

26 June 2003

Produced By:

Department of the Air Force
Air Force Research Laboratory
Rome Research Site
32 Brooks Road
Rome, New York 13441-4114

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

This document specifies the requirements that software applications and tools must meet in order to successfully integrate into existing infrastructures at operational sites. The ability of software programs to meet these requirements is evaluated by the Joint Integration Test Facility (JITF) at Air Force Research Laboratory (AFRL) in Rome, New York and the National Imagery and Mapping Agency (NIMA) Integration Test Facility (ITF) in Newington, Virginia. These two test organizations are designated as Department of Defense Intelligence Information System (DoDIIS) Independent Test Activities and support the DoDIIS Certification Process by verifying that applications integrate into common operational environments.

The common operational environment promoted by DoDIIS emphasizes the objectives of integration, interoperability, and modularity of software applications. The DoDIIS Certification Process has been designed to ensure that applications meet these objectives and critical quality standards for functionality, security, training, interoperability, and integration. The tasking to Program Management Offices (PMOs) and identification of responsibilities for all phases of the Certification Process are specified in the *DoDIIS Instructions 2000.*

Figure 1-1 illustrates the Certification Process defined by the *DoDIIS Instructions 2000* and further described in information provided by the DoDIIS Executive Agent (DExA) for Test and Evaluation (Air Force Command and Control Intelligence Surveillance Reconnaissance Center [AFC2ISRC/A25]).

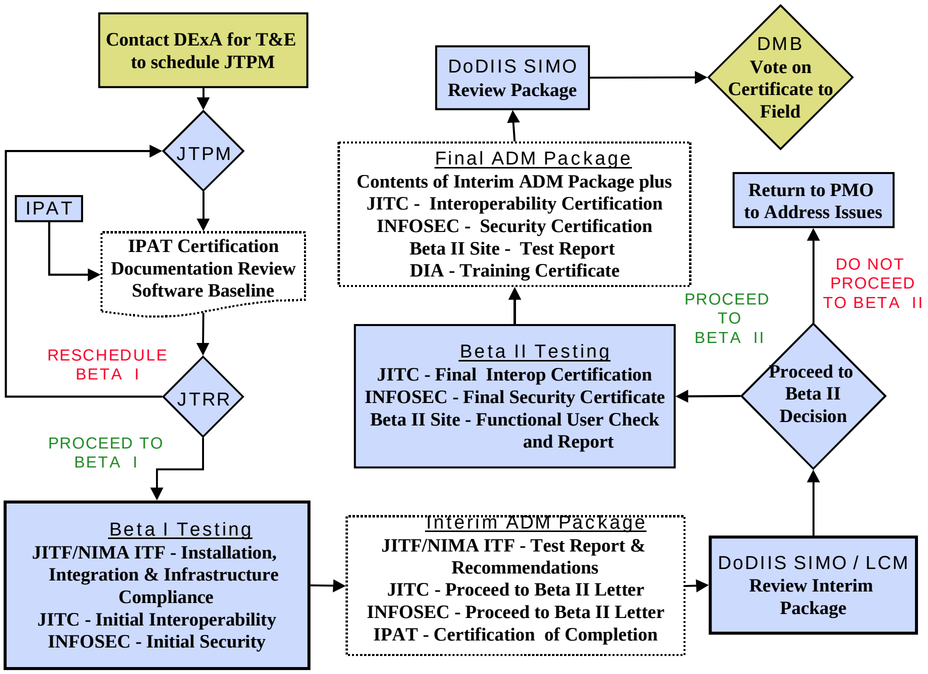
Contact DExA for T&E to schedule JTPM
JTPM
IPAT
IPAT Certification Documentation Review Software Baseline
RESCHEDULE BETA I
JTRR
PROCEED TO BETA I
Beta I Testing
JITF/NIMA ITF - Installation, Integration & Infrastructure Compliance
JITC - Initial Interoperability
INFOSEC - Initial Security
Interim ADM Package
JITF/NIMA ITF - Test Report & Recommendations
JITC - Proceed to Beta II Letter
INFOSEC - Proceed to Beta II Letter
IPAT - Certification of Completion
DoDIIS SIMO / LCM Review Interim Package
Proceed to Beta II Decision
PROCEED TO BETA II
DO NOT PROCEED TO BETA II
Return to PMO to Address Issues
Beta II Testing
JITC - Final Interop Certification
INFOSEC - Final Security Certificate
Beta II Site - Functional User Check and Report
Final ADM Package
Contents of Interim ADM Package plus
JITC - Interoperability Certification
INFOSEC - Security Certification
Beta II Site - Test Report
DIA - Training Certificate
DoDIIS SIMO Review Package
DMB Vote on Certificate to Field

Figure 1-1 DoDIIS Certification Process

The focus of integration testing is to verify that applications function within existing infrastructures and resources. Integration testing verifies installation procedures and infrastructure compliance, identifies computer and network resource conflicts, and the operational impacts of applications cohabiting in a common environment. Integration testing validates that each application will function as a building block of the overall system supporting the Intelligence Community (i.e., DoDIIS).

The integration requirements are derived from infrastructure requirements, technical best practices, and government and industry standards, including the Certified for Microsoft® Windows Application Specifications. The integration requirements are applicable to a broad spectrum of application architectures and consider the dynamic nature of the infrastructure needs of the intelligence community.

The integration requirements contained in this document are organized by category:

- Documentation - These requirements evaluate the content and structure of application documents that the system administrator/installer will rely upon to plan the application’s resource requirements and to determine the effects of the software on the operational and security architectures of the site.
- Configuration and Installation - These requirements evaluate the application installation, configuration, and uninstallation process and the required steps to verify correct installation.
- Environment - These requirements evaluate the operating environment established or required by the application when it begins execution and the potential effects of that environment on other applications.
- Operation - These criteria examine aspects of the application execution that could affect the execution, configuration, or security of other applications, either on the same hardware platform or on other platforms at the site. Included in this category is how administration of the application integrates into the overall system administration strategy of a site.
- User Interface - These criteria are concerned with the integration of the application with the workstation windowing system.
- Integration Security - These requirements identify areas of the design and operation of the application that may affect the site security architecture and the level of effort on the part of system administrators and security officers to maintain the site security architecture. These requirements may address areas of system security architecture that are not identified in the application security documentation.

The integration requirements address integration of applications into client-server operating environments and web-based multi-tiered operating environments. For this reason, a PMO may find that some requirements will not apply to the application because they are designed for one environment or the other. In keeping with the current test process, all requirements will be reviewed for applicability for each test. Software versions will be evaluated against only those requirements that are applicable.

1.1 DOCUMENT ORGANIZATION

This document is organized into the following sections:

Section 1 provides an introduction to integration requirements and additional information.

Section 2 provides a list of references.

Section 3 contains Integration Requirements, including explanations and test methods.

Section 4 describes the JITF process for analyzing the effects of operating system (OS) patches and advisories on the infrastructure.

Section 5 contains a list of acronyms.

Section 6 contains a list of terms and definitions.

1.2 DOCUMENT ACCESS AND UPDATE INFORMATION

Comments and recommendations for changes to this document can be submitted by any reader and should be provided in writing. Please identify the page and paragraph associated with each comment. All written comments will be reviewed and a disposition for each comment will be provided to the originator of the comment. Comments can be submitted via the following means:

U.S. Mail:

Information Management Services (IMS)
RL/IFEB
32 Brooks Rd
Rome, NY 13441-4114

Electronic Mail: cubic_cm@rl.af.mil

Additional copies of this document can be downloaded from the World Wide Web or Intelink at the following addresses:

Internet World Wide Web: http://www.rl.af.mil/jitf/

Intelink: http://web1.rome.ic.gov/vtf.

NIMA program personnel with comments should direct their suggested changes to:

NIMA ITF
Attention: ITF Site Lead
8510 Cinderbed Road
Newington, VA 22102

1.3 INTEGRATION CERTIFICATION CRITERIA

In accordance with the *DoDIIS Instructions 2000*, the JITF is tasked to make "go/no go" recommendations on applications to the DoDIIS Management Board (DMB) as a result of integration testing. This responsibility has also been assigned to the NIMA ITF for NIMA applications. An application will receive a recommendation to proceed if:

- seventy-five percent (75%) of the applicable integration requirements have been met and,

- there are no open Impact Code 1 document or software test findings. Paragraph 1.4.1 defines Impact Code levels.

A "no go" recommendation indicates that there are findings for the application under test that seriously affect the capability of the application to install and/or operate in a site environment without affecting other applications or site operations. The DMB is the decision authority for the Certification Process and uses the JITF or NIMA ITF recommendation in making a final determination for the application to proceed to the next phase.

1.3.1 JITF and NIMA ITF Support to PMOs

The JITF and NIMA ITF are committed to assisting PMOs successfully meet integration requirements and the integration certification criteria. The test organizations' objectives are to expedite the fielding of quality software and function as a resource to the PMOs to ensure that integration is built into software. In keeping with this approach, the following steps have been taken to simplify integration testing for PMOs:

- The JITF and NIMA ITF have representatives responsible for assisting PMOs in understanding the testing process, criteria, and requirements. These individuals can be contacted at anytime or discussion can be initiated by contacting the DExA for T&E. Current Point of Contact (POC) information for JITF personnel is available on the JITF web sites.

- The DExA conducts a Joint Test Planning Meeting (JTPM) approximately three months prior to integration testing to be accomplished by the JITF. The DExA for T&E is invited to participate in NIMA JTPMs. PMOs and their developing contractors, the test organizations, and others as appropriate discuss test objectives and methods in detail.

- Information packages containing detailed information on entrance and exit criteria, pass/fail criteria, checklists, self-assessments, and planning support tools are distributed at the JTPM. These are also available to any PMO upon request.

- A copy of the JITF Automated Tool (JAT) is made available to PMOs. The JAT collects data to support the evaluation of multiple integration requirements and can be used by the PMO as part of its self-assessment.

- The integration requirements and test procedures used to evaluate compliance are fully documented and provided to PMOs in advance of testing. Integration testing is an "open book" test.

- The JITF and NIMA ITF will make every effort to ensure that resources are available to conduct "pre-test" evaluations for PMOs at their request. Pre-test evaluations can be accomplished during Factory Acceptance Testing (FAT) or earlier in the development life-cycle. Pre-test evaluations are strongly encouraged, especially for high risk programs or programs undergoing certification for the first time.

- Integration test engineers are available to answer questions from development contractors and PMOs. PMOs can contact JITF representatives to have their questions and concerns forwarded to an appropriate test engineer for resolution.

1.4 INTEGRATION TEST REPORTS

Test reports are available on the Virtual Test Folder (VTF) that is maintained by the JITF. The VTF is located on Intelink at http://web1.rome.ic.gov/vtf. Registering with the VTF provides interested individuals automated e-mail notification of the availability of test information and documentation. The VTF provides access to interoperability test reports, JTPM memos, JITF OS patch assessments, miscellaneous test items, and integration test reports.

The integration test report, produced by the JITF or NIMA ITF as appropriate, details the extent of compliance with the integration requirements and provides an assessment of the consequences of the resulting level of integration quality of the application. The integration test report includes:

- Evaluation of compliance with the integration requirements, including a detailed listing of document and software findings and their associated impact codes. See paragraph 1.4.1 for a description of impact codes.
- Assessment of effects of non-compliance with integration requirements
- Identification and assessment of other issues that affect the usability of the system baseline in operational environments
- "Go/no go" recommendation for the continued progress of the application through the certification process

1.4.1 Impact Code Levels

The JITF and NIMA ITF evaluate the extent to which the application under test meets each requirement. For each requirement not met by the mission application, the JITF/NIMA ITF generates a test finding and assigns an Impact Code level for that finding. The impact code is a measure of the significance of the finding with respect to integrating the application into site architecture. It is an assessment of the effect of the unmet requirement on the application, existing applications, or the infrastructure and the level of effort required by system administrators and site personnel to exploit the functionality provided by the application. In addition, the design of the application will also influence the significance of requirements that are not met.

Integration requirements do not have equal weight. Some of the requirements deal with areas of the application that are more critical than other requirements. As a result, the JITF and NIMA ITF employ a grading system for test findings which assigns a value to for each impact code. Using this system, an application is scored more accurately, by providing higher scores to test events resulting in a majority of lower category three and four impact codes and lower scores to test events resulting in a majority higher category one and two impact codes.

The following codes are used by integration test teams to indicate the severity or significance of each integration finding.

Impact Code 1

A finding that,

a) identifies baseline adjustments, not included in the installation guide, made during the test event in order to successfully install the application;
b) has a serious effect on the operation of either the application or on another application or component of the infrastructure; or
c) impacts cost, schedule, performance or Post-Deployment Software Support (PDSS).
d) identifies a security vulnerability in the application or site architecture that can be exploited by a general user; or
e) seriously increases the level of effort required by site personnel to manage the application or other applications.

An Impact Code 1 finding is assigned if the application baseline must be changed in order to continue testing, if the installation documentation is not detailed enough to support the successful installation of the application, or if a security vulnerability exists.

The level of effort is a key determinant for Impact Code 1 findings. The time or expertise that is required to install or manage the application cannot exceed what is reasonably expected for an application. For example, if the installation guide says that the application can be installed in a single day, but the installation takes more than 20 working hours, then an Impact Code 1 finding would be generated.

An application cannot proceed to Beta II testing until all Impact Code 1 findings have been resolved by the PMO and verified by the JITF Engineers.

Impact Code 2

A finding that,

a) has a significant effect upon, but does not prevent, the successful installation of the application under evaluation;
b) has a significant effect on the operation of either the application or on another application or component of the infrastructure;
c) impacts cost, schedule, performance or Post-Deployment Software Support (PDSS).
d) creates a security vulnerability in the application; or
e) significantly increases the level of effort required by site personnel to manage the application or other applications.

An Impact Code 2 finding can be resolved by a change in procedure or configuration. The resolution of an Impact Code 2 finding requires a significant level of effort by site administrators. The resolution does not cause significant delay in integration testing;

instead, it can be proposed and evaluated during integration testing at the JITF or NIMA ITF.

Impact Code 2 findings do not cause integration test failure, but the accumulation of Impact Code 2 findings may affect the test organization's "go/no go" recommendation.

Impact Code 3
A finding that,

a) has an effect upon the installation of the application under evaluation;
b) has a effect on the operation of either the application or on another application or component of the infrastructure; or
c) increases the level of effort required by site personnel to manage the application or other applications, but does not require a significant level of effort by site administrators.

The successful resolution of an Impact Code 3 finding requires technical expertise expected of site administrators. The resolution does not cause significant delay in integration testing; instead, it can be proposed and evaluated during integration testing at the JITF or NIMA ITF.

Impact Code 3 findings do not cause integration test failure, but the accumulation of Impact Code 3 findings may affect the test organization's "go/no go" recommendation.

Impact Code 4
A finding that,

a) has little or no effect upon the installation of the application under evaluation;
b) has a little effect on the operation of either the application or on another application or component of the infrastructure; or
c) nominally increases the level of effort required by site personnel to manage the application or other applications, but does not require a significant level of effort by site administrators.

The finding can be resolved by a workaround that can be implemented as a change in during integration testing without a significant level of effort, or the finding can be left as is. Even though the finding has some effect on the configuration or operation of the mission application, or on other components of the site architecture the administrator will be able to manage the mission application.

2. REFERENCES

DOD Directive 5000.1, Defense Acquisition, October 23, 2000, including Change 1, dated January 4, 2001

DOD Directive 5000.2, Operation of the Defense Acquisition System, October 23, 2000

ASD (C3I), *Department of Defense Joint Technical Architecture*, Version 4.0, 2 April 2001

AIA 497th Information Operations Group /INDS, *Test and Evaluation Policy for Department of Defense Intelligence Information System (DoDIIS) Intelligence Mission Applications (*IMA*)*, April 1999

DoDIIS Management Board, *DoDIIS Profile of the DoD Joint Technical Architecture (JTA) and Defense Information Infrastructure Common Operating Environment (DII COE) Version 3.1,* September 2000

DoDIIS Management Board, *DoDIIS Instructions 2000*, February 2000

Protecting Sensitive Compartmented Information Within Information Systems (DCID 6/3)-Manual, 1999

DIA, *Joint DoDIIS/Cryptologic SCI Information Systems Security Standards*, 31 March 2001 Revision 2

Microsoft Corporation, *Designed for Microsoft® Windows NT® 4.0 and Windows® 98 Logo, Handbook for Software Applications,* Version 3.0d, February 4, 1999

Microsoft Corporation, *Application Specification for Microsoft® Windows® 2000* v1.3 September 26, 2000

Microsoft Corporation, *Designed for Microsoft Windows XP Application Specification Microsoft®* Version 2.3 January 2, 2002

AFRL, Common User Baseline for the Intelligence Community (CUBIC) *Configuration Management Plan*, August 7, 2001

National Security Agency, Guide to Securing Microsoft Windows 2000 File and Disk Resources, April 19, 2001

National Security Agency, Guide to Securing Microsoft Windows 2000 Active Directory, December 2000

National Security Agency, Guide to Securing Microsoft Windows 2000 Group Policy: Security Configuration Tool Set, May 17, 2001

National Security Agency, Guide to Securing Microsoft Windows 2000 Group Policy, September 13, 2001

National Security Agency, Guide to Securing Microsoft Windows 2000 Terminal Services, July 2, 2001

National Security Agency, Guide to the Security Configuration and Administration of Microsoft ISA Server 2000, January 7, 2002

National Security Agency, Guide to Securing Windows NT/9x Clients in a Windows 2000 Network, January 23, 2001

National Security Agency, Guide to Securing Microsoft Windows 2000 Schema, March 6, 2001

National Security Agency, Guide to the Secure Configuration and Administration of Microsoft Windows 2000 Certificate Services, October 2001

National Security Agency, Guide to the Secure Configuration and Administration of Microsoft Windows 2000 Certificate Services (Checklist Format), October 10, 2001

National Security Agency, Guide to the Secure Configuration and Administration of Microsoft Internet Information Services 5.0, August 20, 2001

Copies of these materials may be obtained by contacting Information Management Services (IMS), formerly known as CUBIC Configuration Management (CM). Point of contact information is listed in this document under Section 1.2.

3. INTEGRATION REQUIREMENTS

Requirements for integration are provided in this section. A unique identification number is assigned to each requirement for traceability. In addition, the following information is provided for each requirement:

- an explanation/clarification of the requirement
- the evaluation method used,
- impact code ranges associated with the requirement,
- an asterisk “*” attached to the requirement number indicates that the JAT is used to evaluate the requirement, and
- a “#”attached to the requirement number indicates that the requirement is applicable to applications using a centralized server at one location managed by the PMO and accessed via commercially available web browsers. These requirements are also applicable, as appropriate, to all applications undergoing integration testing. If a requirement does not have a “#” attached, it is not applicable to centralized server configurations.
- a “*”attached to the requirement number indicates that the requirement is NOT applicable to NIMA applications.

Each requirement is reviewed for applicability for the version of software under evaluation. Microsoft® OS requirements are evaluated using Logo Testing procedures, which are enhanced where appropriate. Additional Solaris-specific analysis is provided via the Sun Microsystems’ application certification binary compatibility tool.

3.1 DOCUMENTATION

DOC-1# Application documents shall contain page numbers for all sections and appendices.

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION
Page numbering improves the utility of each application document. This can be especially significant when the reader must identify to a third party (such as a help desk) an entry in a document that either has errors or is unclear. Page numbers within a single

document shall not be repeated or skipped.
TEST METHOD
Application documents will be inspected for inclusion of page numbers.

DOC-2[#] Application documents shall contain numbered sections.

IMPACT CODE RANGE: 3-4

REQUIREMENT CLARIFICATION
Construction of a document in numbered sections improves the utility of the document and aids the reader in identifying areas with errors or requiring clarification.
TEST METHOD
Application documents will be inspected for inclusion of numbered sections.

DOC-3[#] Figures and tables in application documents shall have titles and reference numbers.

IMPACT CODE RANGE: 3-4

REQUIREMENT CLARIFICATION
Assigning titles and reference numbers to all figures and tables improves the utility and readability of the document.
TEST METHOD
Application documents will be inspected for inclusion of titles and reference numbers on all figures and tables.

DOC-4[#]* Soft copy documents shall match hard copy versions in content, structure, and sectioning.

IMPACT CODE RANGE: 3-4

REQUIREMENT CLARIFICATION
In order to avoid confusion that may occur when matching a soft copy version of a document to a hard copy version (e.g., when discussing a problem with the application help desk), the two versions should match exactly. At a minimum, the content, structure,

and sectioning of the document should be consistent for both versions.
TEST METHOD
The soft copy version will be compared to the hard copy version. This requirement is met if the content, structure, and sectioning of the soft copy document match the sectioning of the hard copy document. This requirement is Not Applicable if only one type (either soft or hard copy) of documentation is provided.

DOC-5 Application configuration and installation information shall be consolidated into a single configuration and installation document.

IMPACT CODE RANGE: 2-3

REQUIREMENT CLARIFICATION
The application administrator/installer must be able to find all necessary information for the installation of the application in a single, logically ordered, document. This approach lowers the probability of errors during the configuration and installation process. If configuration and installation instructions must be spread beyond a single document, then these documents must specifically reference the parts needed in each other, preferably by section and/or step. If referencing another document, it must be by specific identifier (such as title and date, document reference number, etc).
TEST METHOD
The requirement will be evaluated by inspection of the configuration and installation guide. This requirement is not met if the configuration and installation information is spread across several documents and the references to additional documents are not explicitly stated.

DOC-6 The application documentation shall include installation verification information.

IMPACT CODE RANGE: 2-3

REQUIREMENT CLARIFICATION

Configuration and installation of the application can directly affect the operating and security architectures of the application and of the site. The JITF/NIMA ITF will confirm that the application was successfully installed and configured according to the application baseline. Verification documentation assists the JITF/NIMA ITF, as it would a user site, with this confirmation.

The installation verification documentation should be a subset of the System Test Plan and Procedures, System Security Test Plan and Procedures, Site Acceptance Test Plan and Procedures, or similar documents. It should give the installer confidence that the application has been installed correctly, but should not be an exhaustive functional exercise.

TEST METHOD

Application documentation will be inspected for the inclusion of verification procedures.

The requirement is met if adequate verification documentation is provided. Installation Verification procedures must exercise the main functions of the application to verify that all required software components have been installed and configured correctly.

For example: verification procedures for an imaging application might include:

- querying an image
- requesting the image be sent
- viewing the image
- storing the image

Testing every zoom setting for a viewed image is the domain of functional testing and is not necessary for verifying the installation and configuration.

Adequate verification documentation also covers the steps to execute each main function. The instructions must assume the person conducting the verification is not a knowledgeable user of the application. The application's installer may be the site's administrator who installs multiple applications at the site, but is not a trained user of the application. For example, it would not be acceptable to say "Query an image" without stating the steps to do so with that application.

**DOC-7 ***The application configuration and installation guide shall specify if the application requires a dedicated platform for the application server or if the application server can be installed on a platform shared with other application servers.

IMPACT CODE RANGE: 2-3

REQUIREMENT CLARIFICATION

One goal of the common infrastructure is to give sites flexibility in selecting how each application will be installed and used. An application that, by design, permits sharing of a platform with other application servers allows sites to select platforms based upon application performance and resource usage. An application that, by design, requires a dedicated platform may hinder integration of the application into a site simply because the site is forced to acquire and install hardware and extend its application administration strategy to cover the new application. Additionally, dedicated servers should be avoided where mirroring of the application's server is required to insure 24/7 availability and disaster recovery. Sites may not have the resources to dedicate a second server platform solely for the use of one application.

Application servers can include Storage Area Networks (SANs) and Network Attached Storage (NAS). SANs and NAS involve connecting storage to servers over networks. Support for SANS or NAS is not currently required, but must be fully documented when supported. SANS and NAS are expected to become more prevalent as bandwidths increase and protocols become standardized.

There are risks associated with both approaches. The extent of the risk with regard to site integration depends upon the quality of the application configuration and installation guide and on availability of resources and personnel to install and manage the application.

TEST METHOD

Application configuration and installation guide will be inspected to verify that the need for a dedicated server platform or the ability to share a server platform is specified.

The absence of this information results in an assessment of Does Not Meet.

DOC-8 The application installation and configuration guide shall contain step by step instructions to perform application installation and configuration.

IMPACT CODE RANGE: 1-4

REQUIREMENT CLARIFICATION

The goal of application configuration and installation guide is to permit the reader (e.g., the application administrator) to install and configure the application without error. The configuration and installation guide should not increase the probability of error due to lack of clarity or information.

TEST METHOD

Installation and configuration guide will be inspected for step by step instructions. Each step should be concise and constitute a single action. The step should be explained sufficiently to avoid unnecessary guesswork or presumptive decisions by the installer.

The requirement is not met if the installation is not written in step by step format, if one or more steps are missing, or are sufficiently unclear that the installer can not decide how to proceed.

**DOC-9 ***The application configuration and installation guide shall include instructions to add the application to the infrastructure application selection mechanism.

IMPACT CODE RANGE: 2-3

REQUIREMENT CLARIFICATION

The installation process must include the steps to add the application to the application selection mechanism (e.g., background window menu, application folder, etc.). The installation procedure provided by the application developer must include the application name, executable location, and the command lines that are required to set needed environment variables and launch the application.

TEST METHOD

The application configuration and installation guide will be examined to verify that instructions for adding the application to the infrastructure application selection mechanism are included. Once the installation has been completed, the application selection mechanism (e.g., background window menu) will be invoked on the test workstation. Verify that an entry for the application appears in the menu as documented in the installation procedures. Select the application from the background menu and verify the execution of the application.

Automatic addition of the application to the infrastructure application selection mechanism is acceptable.
This requirement is Not Applicable if the application is run within a web browser.

DOC-10# Application documentation shall specify points of contact (phone, electronic mail, etc) for application support.

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION
Administrators and users must be able to identify and communicate with personnel who can assist with questions and problems. This information must be contained in the appropriate application documentation. Telephone and electronic mail are acceptable forms of communication.
TEST METHOD
Application documents will be inspected to verify that points of contact are provided. The information must include the office or organization name, telephone number (s), and electronic mail address, if one is available.

DOC-11 The application configuration and installation guide shall specify the minimum amount of disk space needed to install and execute the application.

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION
All space requirements and specific file systems, if any, needed to install and run the application must be specified. This includes disk space for executables, as well as storage for application and user data.
TEST METHOD
Configuration and installation guide will be inspected to verify that minimum disk space is specified.

DOC-12 Not applicable for Version 3.0 and above test procedures. Incorporated into DOC-11.

DOC-13 The application configuration and installation guide shall specify the recommended size of random access memory (RAM) required to execute the application.

IMPACT CODE RANGE: 2-3

REQUIREMENT CLARIFICATION
This is typically a performance issue; applications should make recommendations on RAM for site consideration. This specification

should be made for both user workstations and application server platforms.
TEST METHOD
Configuration and installation guide will be inspected to verify that RAM size is recommended.

DOC-14 The application configuration and installation guide shall specify the OS versions and OS packages/subsets that must be installed to support the application.

IMPACT CODE RANGE: 2-3

REQUIREMENT CLARIFICATION
The application should not require that each site install the full OS load as routine practice. Therefore, the application should identify the software dependencies with regard to specific OS version and also the OS modules (i.e., subset packages or resource kits) that must be installed in order for the application to operate properly.
TEST METHOD
Configuration and installation guide will be inspected to verify that OS versions and packages/subsets/resource kits are specified. The absence of this information results in an assessment of Does Not Meet.

DOC-15 The application configuration and installation guide shall specify the OS patch levels that must be installed to support the application.

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION
Application developers make independent decisions regarding patch level compatibility. Therefore, the Configuration and installation guide must state known dependencies upon patch levels. This may not be a significant issue for sites that stay current with all OS packages. However, it is necessary information for sites that may not be current and is an incentive for site administrators to update patch levels on site workstations. The documentation shall include information as to what OS patches may be required.

TEST METHOD

Configuration and installation guide will be inspected to verify that patch levels for each supported OS are specified.

For Microsoft OS platforms: include required service packs/hotfixes.

The requirement is met if the specific patch list is provided; it is not sufficient to simply require “the latest patches”.

DOC-16 The application configuration and installation guide shall specify any modifications made to the OS configuration that are required to support the application.

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION

Modifications to the UNIX kernel or to the Microsoft OS configuration are not necessary for most applications. Modification would be required if the application requires an additional hardware device, additional software resources such as interprocess communication, or additional drivers for I/O devices. In such situations, the necessary modifications must be clearly stated in the configuration and installation documentation.

Kernel software is divided into groups of related functions called *modules*. Some modules provide platform-specific operations. Other modules are device drivers. Device drivers are dynamic kernel modules that are loaded when the device is accessed.

The kernel modules are stored in three directories-two under the root file system and one under the /usr file system:

/platform*/sparc/kernel for SPARC platforms*
/kernel-Common kernel modules required for booting
/usr/kernel-Common kernel modules used by platforms with a particular instruction set

The /etc/system file is used to determine which kernel modules are loaded and to define various kernel parameters.

TEST METHOD

Configuration and installation guide will be inspected to verify that modifications for each supported OS are specified.

This requirement is Not Applicable if no modifications are required.

DOC-17 The application configuration and installation guide shall specify additional hardware and associated drivers that are required to support the application.

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION
If the application requires additional hardware and installation of software drivers to control the hardware, the configuration and installation guide will clearly specify the steps needed to successfully install and configure both.
TEST METHOD
Configuration and installation guide will be inspected to verify that instructions to install additional hardware and associated software drivers in each supported OS are specified. If no additional hardware and installation of software drivers to control the hardware are utilized, this requirement is Not Applicable.

DOC-18 The application configuration and installation guide shall specify additions/modifications to system configuration files that are required to support the application.

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION
Many applications may use system configuration files. Because these files are a shared resource no application should make undocumented changes to them. In addition, the application installation process must not overwrite system configuration files. Information that was added by other applications may be lost. Instead, the application should add entries to the existing files and include the pertinent details in the application installation and configuration documentation. Undocumented changes to system configuration files may cause conflict within the computing environment. System administrators need to be aware of all configuration changes in order to avoid such conflicts and manage and maintain reliable information processing capabilities.

TEST METHOD
Review the configuration and installation guide to verify that all modifications to system configuration files are specified. For UNIX, review files such as /etc/hosts, /etc/services, and /etc/syslog.conf to verify that they have not been overwritten, and that any changes or modifications have been documented. For Microsoft OSs, documentation must clearly specify the settings for computer peripherals that are required by the application. No undocumented changes to the Microsoft OSs Registry, Windows.ini, System.ini, Config.sys, or Autoexec.bat files shall be made.

DOC-19 The application configuration and installation guide shall provide rules defining appropriate file ownerships and permissions for all files and directories that are loaded or modified during application installation.

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION
Application documentation should include information on file ownerships and permissions. This is needed to permit the security officer or administrator to confirm that all ownerships and permissions are set correctly during installation. The information must be included even if the installation is completely automated.
TEST METHOD
The appropriate application documentation, e.g., Configuration and Installation Guide, Version Description Document (VDD), will be examined for the inclusion of file ownerships and permissions for all files created or modified during configuration and installation of the application.

DOC-20 The application configuration and installation guide shall specify the audit configurations (i.e., audit flags, etc.) that must be set in order to meet the application security requirements.

IMPACT CODE RANGE: 2-3

REQUIREMENT CLARIFICATION
DoDIIS security policy permits applications to rely on the underlying OS audit function for auditing of application activity. For such

applications, the Configuration and Installation guide must clearly specify the audit flags that must be set in order to meet the application's security concept of operations. If an application does not rely on any auditing by the underlying OS, then the application documentation should clearly state that no specific settings are required.

TEST METHOD

Configuration and installation guide will be inspected to verify that audit flags for each supported OS are specified.

This requirement is met if the audit flags are specified.

DOC-21 The application configuration and installation guide shall identify other software products on which the operation of the application is dependent.

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION

Even simple applications may depend upon the presence and operation of third party software. This typically is true for applications that rely on database management systems, word processing systems, or on shareware software that is integrated into the IMA baseline. In each case where the application depends upon the presence and operation of third party software, IMA documentation, such as the Configuration and Installation Guide or Version Description Document, will clearly state the identity of the software, the version and patch level of the software, and the nature of the dependency. This includes specification of all shareware products in the IMA baseline, including those used only to install or uninstall the IMA.

TEST METHOD

Application configuration and installation guide will state the name, version, and patch level of other software on which the application depends. The nature of each dependency will be stated.

Once the application is installed, the application directory tree will be scanned to identify all shareware software. The listing generated by the scan will be compared to the listing of shareware products provided in the IMA documentation. If there is shareware found, and no reference is made in the documentation, this requirement is not met.

The requirement is met if no dependencies exist.

DOC-22 Comprehensive instructions shall be provided for uninstalling the application, including backing out of a failed installation so that it can be reinstalled.

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION
Operator errors or script problems may cause the application installation to fail and thus require a partial or total rollback of the installation. Application installation should not be like a black box with respect to determining exactly which portions may have been installed before a failure occurred. Additionally, the initial point of failure may not be detected. This means the installation may continue even after part of the installation has failed. The error may be discovered, or the whole installation may fail. During this time, additional undetected errors may occur as consequences of the original error. The residue left from the failed attempt may cause conflicts during the next installation attempt. Without instructions to back out of the installation, the only way to fully insure a clean reinstallation may be to install the entire application from the OS up. This should be avoided. The installation and rollback strategy should be designed so that the installation would only be rolled back to the point of failure or to the beginning of the segment or module where the error occurred.
TEST METHOD
The requirement will be met by inclusion of rollback instructions in the configuration and installation documentation.

DOC-23[#] Application documentation shall specify the browsers and browser versions that are compatible with the application.

IMPACT CODE RANGE: 2-3

REQUIREMENT CLARIFICATION
Applications should test against browser versions that are currently in use in the community (i.e., not only the latest versions). The application documentation should state which browsers are known to be compatible with the application.
TEST METHOD
Application documentation will be inspected to verify that compatible browsers are identified.

This requirement is Not Applicable if the application does not use a browser.

DOC-24$^{\#}$ The application configuration and installation guide shall specify any browser settings that are necessary to access the application.

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION
Applications should not assume specific browser settings because site policy may dictate browser configuration. However, if there are configuration settings that are necessary (e.g., Java enabled), the Configuration and Installation Guide must identify them. If additional viewer software is required, the document should include a source, preferably Intelink Central, and configuration information.
TEST METHOD
Application documentation will be inspected to verify that necessary browser settings are identified. If additional viewers are required, the documentation should include information including, but not limited to, a source for the software, MIME type, and filename extensions to be used. This requirement is Not Applicable if the application does not use a browser.

DOC-25$^{\#}$ If the application design requires the use of plug-ins, the application documentation shall include a list of required browser plug-ins, the compatible versions of the plug-ins, the source of the plug-ins, and appropriate licenses.

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION
Since access to sources for browser plug-ins is extremely limited on classified networks, the administrator or user must be notified before the application is used that a plug-in is necessary. Therefore, the configuration and installation guide must list the plug-ins that are required and how the plug-ins and licenses (if required) can be obtained. In addition, the software should be submitted to the ISMC for inclusion in the Intelink download archive. Downloading and

installation of software obtained from unclassified sites is discouraged on classified systems.

TEST METHOD

Application documentation will be inspected to identify the required plug-ins and a classified source for each plug-in, along with information indicating which Java Plug-in versions are compatible with the application.

The documentation must also include instructions to install and configure the plug-ins. In most cases, configuration and installation is performed automatically by the browser; any additional manual steps must be included in the documentation.

This requirement is Not Applicable if the application does not require plug-ins.

DOC-26[#]*Application documentation shall include identification of application server and Java applets registration with Intelink Central if Java applets are used.

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION

DODIIS policy states that Java applets must be registered with Intelink Central and that a code review of each applet should be conducted. Intelink policy states that only registered applets are permitted on servers accessible through Intelink.
The *DoDIIS Instructions* do not specifically state which organization is responsible for reviewing Java applet source code. The code review can be done by the security certifiers or a third party organization. It is the responsibility of the PMO to arrange code review.

NOTE: DOC-26 verbiage has been updated per the memorandum dated 24Jan2001

TEST METHOD

The application documentation will be inspected to determine if Java applets are implemented.
Additionally, the application software will be checked for any undocumented Java applets.
Java applets can be found by looking for “APPLET” tags within the application’s HTML files.

For UNIX:

cd to the directory where the HTML files are (e.g. “htdocs” directory)

```
# find . –depth ( -name “*.htm*” –o –name “*.HTM*”)
-a –exec grep –I applet {} “;”
```

For Microsoft OSs:

On the Taskbar select “Start” -> “Find” -> “Files or Folders”.
A “Find” window will appear.
In the “Look in:” box of the “Find” window enter the path to the directory where the HTML files are located.
Select the “Advanced” tab.
In the “Containing text:” box enter “applet”.
Select the “Find Now” button.

Java applets may be hosted only on servers that are registered with Intelink Central. The server registration process does not produce written confirmation. Proof of registration is demonstrated by the listing of the mission application server on the Intelink Central Home Page. The registration of Java applets can be done on-line with Intelink Central. Copies of the registration forms can be included with the mission application documentation as documentation of registration.

If the application documentation does not include proof of registration, the JITF/NIMA ITF test engineers will review the applet registration pages on the Intelink Central Home Page. The requirement is not met if the applet(s) is not registered.

Documentation of applet code review must include the date of the review, name and address of the reviewer(s), and a summary of findings and resolutions from the review.

This requirement is Not Applicable if the application does not use Java applets.

DOC-27 Not applicable for Version 3.0 and above test procedures.

DOC-28 Not applicable for Version 4.0 and above test procedures. Incorporated into INST-28.

DOC-29 Not applicable for Version 3.0 and above test procedures.

DOC-30$^{\#}$ Application installation and configuration documentation shall identify the use of DoDIIS standard products in accordance with the *DoDIIS Profile of the DoD Joint Technical Architecture* (JTA).

IMPACT CODE RANGE: 1-4

REQUIREMENT CLARIFICATION

The DMB publishes the *DoDIIS Profile of the DoD Joint Technical Architecture (JTA)* to maintain continuity between DoDIIS and DoD direction with respect to technical and system architecture specifications. The DODIIS Profile identifies information technologies and software products that will be used in applications fielded at user sites. It defines the community baseline for commonly used support tools such as; browsers, viewers, and database front ends, and infrastructure components, such as OSs and database management systems. The objective is to provide commonality and consistency among application development and integration activities and site configuration activities, reducing the need to maintain multiple baselines of commercial and Government developed products at user sites. The *DoDIIS Profile* refines and interprets the *DoD JTA* guidance in areas where that document is open to interpretation.

The JITF/NIMA ITF supports enforcement of the policies stated in the *DoDIIS Profile* by verifying that products specified in the DoDIIS Profile are used by applications that require the services of those products.

TEST METHOD

The JITF/NIMA ITF will review the application Work Plan and the application Installation and Configuration Guide to identify documentation of Commercial Off The Shelf (COTS), Government Off The Shelf (GOTS), and shareware products that are integrated into the application. For each product, the JITF/NIMA ITF will identify the service that is provided by the product and verify that the product is included in the product matrix provided in the *DoDIIS Profile*.

This requirement is met if the application is compliant with the DoDIIS Profile and supplies supporting documentation.

DOC-31 Application administration documentation shall identify locations of log files, temporary files, and audit data. (UNIX and Microsoft OSs).

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION
Identifying the location of log files, temporary files, and audit data is essential to the maintenance and administration of the application. The application may use the syslog file, temporary directory, and audit directories provided by the infrastructure. Data base Management System (DBMS) transaction logs are also covered by this requirement. Regardless of location, the application administration documentation should clearly identify them.
TEST METHOD
Application administrative documentation shall be examined to determine if the file locations are clearly identified.

DOC-32 The typographical conventions and syntax in application documentation shall be uniform and grammatically correct.

IMPACT CODE RANGE: 3-4

REQUIREMENT CLARIFICATION
Typographical conventions, grammer, and syntax must be uniform throughout the documentation. These elements are essential aids to easily understanding the application documentation. Proof reading for Typographical conventions, grammar, and syntax errors is an essential part of the documentation development cycle.
TEST METHOD
Application Documentation will be inspected for uniform and correct typographical conventions, grammar and syntax. For example, the document will be evaluated for uniform naming and typographical conventions, sentence structure, properly set bookmarks in the Table of Contents, and consistent fonts.

3.2 INSTALLATION AND CONFIGURATION

INST-1 Application installation shall not require installation of the OS. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1

REQUIREMENT CLARIFICATION
In accordance with the integration methodology developed by the community, installing the application can and should be done on a previously installed and executing OS. There should be no requirement to reload the OS simply to install another application. Additional packages/ subsets/resource packs can be added to the OSs, and the OS configuration can be modified without requiring a new installation of the OS. Reloading the OS means the rest of the system (i.e., other applications) must be backed up and restored. This is a time consuming process, particularly if many workstations in the site are affected.
TEST METHOD
The requirement is not met if the configuration and installation documentation calls for an OS reload or if the application's configuration and installation scripts reload the OS. If the actual installation of the application cannot be successfully completed without reloading the OS, then the requirement is not met. This requirement does not apply to releases containing OS version upgrades.

INST-2 Not applicable for Version 4.1 and above test procedures. Incorporated into INST-15.

INST-3 Not applicable for Version 4.1 and above test procedures. Incorporated into INST-15.

INST-4 Not applicable for Version 4.2 and above test procedures. Incorporated into INST-15.

INST-5 Application shall support installation on user workstations and on application servers for export to user workstations. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION
One goal of common infrastructure is to permit sites to allocate their computing resources according to their needs rather than according to the design of individual applications. An application should be designed so that a site can install it on individual workstations or on an application server. Application servers can include Storage Area Networks (SANs) and Network Attached Storage (NAS). SANs and NAS involve connecting storage to servers over networks.
TEST METHOD
The application will be loaded on a user workstation. Once the installation is complete, test cases from the application test procedures will be executed to demonstrate the successful execution of the application. The application will be loaded on an application server. The application will be exported for execution by user workstations. Following installation of the application test cases from the application test procedures will be executed to demonstrate execution of the application on user workstations. This requirement is not applicable to web-based applications that require only a browser on a client platform.

INST-6*Application shall not modify or delete the native programming utilities and libraries. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION
In order to increase the portability of applications and to simplify the installation and management of applications, the infrastructure services that are available to applications must be kept stable. Since the infrastructure will provide a common set of services and functions to all applications, an application must not replace or modify parts of the underlying OS or software run-time environment.
TEST METHOD
After configuration and installation of the application, the state (i.e., modification time, ownership, etc.) of the directories containing

programming utilities and libraries will be compared to the state of these same directories before the application was installed. The JAT results directory will be used to expedite searching.

It is not acceptable for the application to install a library that is a duplicate of a system library. On UNIX platforms check the application utilities and library directories by executing the following commands and noting the modification date on each library:

UNIX:

```
sh
# for i in /bin /usr/bin /sbin /usr/sbin /usr/openwin/bin \ /usr/ucb /usr/etc/lib /usr/lib /usr/openwin/lib /etc/lib \ /etc/security/lib
> do
> echo Checking directory $i
> find $i \( –mtime –X –o –ctime –X \) –exec ls –lad {} “;”
> done
```

(where *X* represents time in days [e.g. 3])

Microsoft:

NT

Select “Start” → “Find” → “For Files or Folders …”
In the “Name & Location” tab:
In the “Named” text box enter “.exe”
In the “Look in” pull down selection box select the appropriate drives or directory.
In the “Date Modified” tab:
Select the “Find all files modified:” radio button
Select the “between” radio button
Enter the start and end dates for the application installation.
Select the “Find Now” button
(Repeat steps replacing .exe with .dll)

In the search results window, investigate any modified .dll or .exe files that belong to the operating system or infrastructure.

2000**:**

Select “Start” → “Search” → “For Files or Folders …”
In the “Search for files or folders named:” text box enter “.exe”
In the “Look in” pull down selection box choose the appropriate drive or directory.
Select “Search Options >>”
Select “Date” check box
Select “files Modified” from the pull down text box
Select the “between” radio button
In the “between” text box enter the date the application installation started.
In the “and” text box enter the date the application installation was completed.
Select the “Search Now” button
(Repeat steps replacing .exe with .dll)

In the search results window, investigate any modified .dll or .exe files that belong to the operating system or infrastructure.

XP**:**
Select “Start” → “Search” → “For Files or Folders …”
In the “Search for files or folders named:” text box enter “.exe, .dll”
In the “Look in” pull down selection box choose the appropriate drive or directory.
Select “Search Options >>”
Select “Date” check box
Select “files Modified” from the pull down text box
Select the “between” radio button
In the “between” text box enter the date the application installation started.
In the “and” text box enter the date the application installation was completed.
Select the “Search Now” button
In the search results window, investigate any modified .dll or .exe files that belong to the operating system or infrastructure.

INST-7*The application shall not require modification of networking protocols or services. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION

Since network protocols and services are infrastructure services, they are not “owned” by any application. Therefore, modification of these services is not permitted.

This requirement also covers dependencies of the application on services such as NIS and NIS+ on UNIX platforms. The selection of such a service is a site choice; the application cannot dictate which service the site can use or force the site to modify the network information service configuration of client and server systems. Instead, the application should be designed to operate with either service running or with none running. An application that explicitly requires the use of NIS rather than being capable of operating under NIS or NIS+ will not meet this requirement.

Since an application cannot assume that it has control over the configuration of workstation resources, it cannot modify the default or standard RPC values. This may cause unpredictable behavior on the part of other applications. The application may append additional RPC values that do not conflict with registered RPC values.

TEST METHOD

After configuration and installation of the application, the state (i.e., modification time, ownership, etc.) of the directories containing the networking protocols and services will be compared to the state of these same directories before the application was installed. The networking services are found within the standard application directories.

Unix:
Check to see if inetd is configured to start a process differently from the application process for a given service or if the application has added a new, non-standard service by executing the command:

For NIS+:

```
ls –l /etc/services
```

If the time indicates that the file has been modified during the installation, execute the command:

```
cat  /etc/services
```

Continue by executing the command:

```
cd /var/nis/data  or  cd /var/nis/<hostname>
ls –l services.org_dir.log
```

If the time indicates that the file has been modified during the installation, execute the command:

```
niscat services.org_dir
```

for NIS:

```
ls –l /etc/services
```

If the time indicates that the file has been modified during the installation, execute the command:

```
cat  /etc/services
```

Continue by executing the command:

```
cd /var/yp/src
ls –l services
```

If the time indicates that the file has been modified during the installation, execute the command:

```
ypcat services
```

LOCAL:

```
ls –l /etc/services
```

If the time indicates that the file has been modified during the installation, execute the command:

```
cat  /etc/services
```

On Solaris platforms, verify that the “nsswitch.conf” file has not been altered as a result of the application installation. Compare the contents of the /etc/nsswitch.conf file before installation of the application to /etc/nsswitch.conf after installation. There should be no changes to the file.

Verify that the application design does not require overwriting or replacing the native RPC Map and that the installation of the application does not include overwriting or replacing the native RPC Map.

The contents of the /etc/rpc file and the rpc map will be examined.

NIS+:

```
ls –l /etc/rpc
```

If the time indicates that the file has been modified during the installation, execute the command:

```
cat  /etc/rpc
```

Continue by executing the command:

```
cd /var/nis/data  or  cd /var/nis/<hostname>
ls –l rpc.org_dir.log
```

If the time indicates that the file has been modified during the installation, execute the command:

```
niscat rpc.org_dir
```

<u>NIS:</u>

```
ls –l /etc/rpc
```

If the time indicates that the file has been modified during the installation, execute the command:

```
cat  /etc/rpc
```

Continue by executing the command:

```
cd /var/yp/src
ls –l rpc
```

If the time indicates that the file has been modified during the installation, execute the command:

```
ypcat rpc.bynumber
```

<u>LOCAL:</u>

```
ls –l /etc/rpc
```

If the time indicates that the file has been modified during the installation, execute the command: cat /etc/rpc

Microsoft OSs:

Use the results directory to expedite searching.

NT:

Select “Start” → “Find” → “For Files or Folders …”
In the “Name & Location” tab:
In the “Named” text box enter “services”

In the “Look in” pull down selection box select the appropriate drives or directory.
In the “Date Modified” tab:
Select the “Files modified:” radio button
Select the “between” radio button
Enter the start and end dates for the application installation.
Select the “Find Now” button
(Repeat steps replacing services with protocol)

If any files appear in the results window, review the contents for changes made by the application.

2000:

Select “Start” → “Search” → “For Files or Folders …”
In the “Search for files or folders named:” text box enter “services”
In the “Look in” pull down selection box choose the appropriate drive or directory.
Select “Search Options >>”
Select “Date” check box
Select “files Modified” from the pull down text box
Select the “between” radio button
In the “between” text box enter the date the application installation started.
In the “and” text box enter the date the application installation was completed.
Select the “Search Now” button
(Repeat steps replacing services with protocol)

If any files appear in the results window, review the contents for changes made by the application.

XP:
Select “Start” → “Search” → “For Files or Folders …”
In the “Search for files or folders named:” text box enter “services,protocol”
In the “Look in” pull down selection box choose the appropriate drive or directory.
Select “Search Options >>”

Select “Date” check box
Select “files Modified” from the pull down text box
Select the “between” radio button
In the “between” text box enter the date the application installation started.
In the “and” text box enter the date the application installation was completed.
Select the “Search Now” button
If any files appear in the results window, review the file contents for changes made by the application.

Examine the results directory for modified registry keys and sub keys that may apply to this requirement. Investigate and evaluate modifications. For example, keys to look at may include:
HKEY_LOCAL_MACHINE\SOFTWARE\Description\Microsoft\Rpc
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Rpc
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\RPCLOCATOR
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Root\LEGACY_RPCSS
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RPCLOCATOR
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RPCSS
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services

INST-8 – Not applicable for Version 3.0 and above test procedures. Requirement converted to OPS-26.

INST-9*The application can be uninstalled using instructions provided in application configuration and installation guide. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION
Operator errors or script problems may cause the application installation to fail and thus require a partial or total rollback of the installation. Application installation should not be like a black box with respect to determining exactly which portions may have been installed before a failure occurred. Additionally, the initial point of failure may not be detected. This means the installation may continue even after part of the installation has failed. The error may be discovered, or the whole installation may fail. During this

time, additional undetected errors may occur as consequences of the original error. The residue left from the failed attempt may cause conflicts during the next installation attempt.

Without instructions to back out of the installation, the only way to fully insure a clean reinstallation may be to install the entire application from the OS up. This is a drastic step that should be avoided. The installation and rollback strategy should be designed so that the installation would only be rolled back to the point of failure or to the beginning of the segment or module where the error occurred.

TEST METHOD

During installation of the application, the test engineers will record if the installation creates backup copies of system configuration files that are modified by the installation process.

Configuration and installation of the application will use incorrect data and/or script errors to induce appropriate installation failures. Following the installation failure, the application will be uninstalled using the instructions provided in application documentation.

The requirement is met if the application can be uninstalled successfully, and the installation of the application can be successfully restarted and completed.

If testing time is available and circumstances permit, after the application has been successfully installed, the application will be uninstalled by following the instructions in the application documentation.

The requirement is met if the system is restored to the state existing before the application was initially installed. This includes recovery of all modified files, deletion of any file systems that were created during the application installation, and removal of any system configuration changes that were made during application configuration.

INST-10 The application installer shall not be required to make changes to installation scripts as part of the installation process. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION

Installation scripts are part of the application baseline. Direct installer modification of configuration and installation scripts violates the concept of a frozen software baseline. Applications should be designed for site integration with choices performed by logical operators like “if” and “case” statements instead of requiring the installer to modify the script code at each site. This is especially true for logical choices involving the various OSs supported by the application. If physical changes must be made to the scripts at end sites, the changes should be generated by other code, which is included in the software baseline.

TEST METHOD

The requirement will be verified during configuration and installation of the application.

Changes to any installation scripts that are required for the configuration and installation to be successfully completed will be recorded by the JITF/NIMA ITF. Changes include adding or modifying environment variable declarations, modifying file and directory paths, correcting typographical errors, and modifying script logic.

The requirement is not met if any installation script is opened for editing and any edits are saved.

INST-11 The application installer shall not be required to enter extraneous or unnecessary information during installation. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION

The installer should be prompted to enter only what is necessary.

TEST METHOD

Input that is required during configuration and installation of the application will be necessary and pertinent to the scope of the system.

The requirement is met if all input is judged as relevant to the current use of the software. The requirement is not met if the input refers to non-existent objects or purposes that are not part of the design of the current application.

INST-12 Manual input for configuration and installation shall be limited to responding to prompts and/or editing configuration file(s) and shall not involve entering information that the script can obtain automatically. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION
The application administrator/installer should not be required to enter large amounts of data during the installation process. The installation process should prompt the administrator when input is required, but the amount of information should be kept small in order to lower the probability of input error. Entry of highly technical and product-specific data may increase the difficulty of determining where errors may have occurred during installation. The problem is particularly acute when the commands and data are beyond the knowledge level of the installer. The installation script should not prompt the installer for system or application information that can be obtained automatically. Examples of such information include hostname, addresses, and OS version.
TEST METHOD
Configuration and installation of the application will verify the requirement. The requirement is not met if, during the installation, data must be entered that can be obtained automatically by an installation script. The tester will identify the function or command that can be used to obtain the information.

INST-13$^{\#}$ The initial configuration and installation parameters shall be consistently set across the software components comprising the application. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION
In some cases, inconsistently set parameters are due to a failure to reconcile the parameters between the various modules of the application software. This may happen, for example, when some modules of the application software are redesigned for a new release without examination of the other modules for resulting discrepancies or conflicts. The discrepancies or conflicts may exist in paths (including library paths) and environment variables, as set in various modules of the installation script.
TEST METHOD

Examine installation scripts and identify parameters (e.g., environment variables, path names, configuration settings) that are initialized more than once, even to the same value.

The requirement is not met if the installer must manually set an installation or configuration parameter more than once (e.g., initializing the root directory for the application).

The requirement is not met if the same installation parameter is not initialized with the same value in all cases and must be modified to enable the installation to continue normally.

INST-14*The application shall not reserve an explicit group identifier (ID) or user ID on UNIX platforms or a specific user/group on Microsoft OS platforms. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION

Selection of user and group Ids across the community can be difficult. An application cannot assume that any given ID value or range of ID values is not already in use at a site where the application will be installed. Therefore, it is better to refer to logical user and group names instead of specific ID values. The application configuration and installation document may recommend one or more values for Ids, but if it does so, the documentation should also recognize the possibility of conflicts and include steps to resolve conflicts that do occur.

TEST METHOD

The application configuration and installation guide will be examined for the presence or absence of instructions to add specific IDs for groups or users and users required by the application configuration.

The requirement is not met if the installation guide states a specific user ID or group ID that must be used or if the installation script uses a specific user ID or group ID without providing the administrator the option of selecting one.

INST-15*The Application installation shall only install COTS or GOTS support or stand-alone applications that are approved under the DoDIIS Profile and are versions that do not already exist on the target platform.(UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION

COTS or GOTS loaded by application installations must be included in the DoDIIS Profile.
The installation of the application must not assume or otherwise require reinstallation of currently installed COTS or GOTS applications. Installing a required pre-existing application that differs only in version does not violate this requirement.

Support applications are software that is commonly used by other applications or users. Such software can include word processors, spread sheets, browsers, and file transfer utilities, typically provided by a component of the infrastructure for general use. The application under evaluation can assume that necessary support applications are either present or can be readily installed.

While DoDIIS approved COTS and GOTS may be included on an installation CD to expedite the installation of these products at sites, the choice of whether to install these products must be left up to the site administrator. If the site is currently running a COTS or GOTS product required by the application, adding another copy as part of the application installation is redundant and does not allow for the efficient use of site resources (disk space).

TEST METHOD

The installation process will be monitored for the installation of COTS and GOTS software, including shareware.
Verify that any COTS and GOTS installed in support of the application are approved under the DoDIIS Profile and are versions that do not already exist on the platform.

The JAT results directory may be used to expedite the examination of application files.
Examine appropriate directories to determine if any support applications have been loaded or overwritten.
For each support application that is found, generate a test finding that lists the application and its normal source of availability (e.g., Intelink for a browser utility) so that the application installation will be able to specify where to obtain the application.

The following commands can be used to help locate COTS and GOTS loaded by the application. The commands search for all executable files added during the application's installation. The output from the commands should be manually inspected for any executables related to associated COTS and GOTS.

UNIX: find / -ctime -*n* | xargs file | grep "executable SPARC" | cut -f1 -d: | xargs ls –acld (Where *n* is the number of days ago the application installation was started.)

Microsoft:

NT:

Select “Start” → “Find” → “For Files or Folders …”
In the “Name & Location” tab:
In the “Named” text box enter “.exe”
I n the “Look in” pull down selection box select all hard drives on the system”
In the “Date Modified” tab:
Select the “Find all files created or modified:” radio button
Select the “between” radio button
In the “between” text box enter the date the application installation started.
In the “and” text box enter the date the application installation was completed.
Select the “Find Now” button
(Repeat above steps looking for “.com” and “.dll” in place of “.exe”)

2000:

Select “Start” → “Search” → “For Files or Folders …”
In the “Search for files or folders named:” text box enter “.exe”
In the “Look in” pull down selection box select all hard drives on the system”
Select “Search Options >>”
Select “Date” check box
Select “files Created” from the pull down text box
Select the “between” radio button
In the “between” text box enter the date the application installation started.
In the “and” text box enter the date the application installation was completed.
Select the “Search Now” button

(Repeat above steps looking for “.com” and “.dll” in place of “.exe”)

XP:

Select “Start” → “Search”
or
Select “Start” → “Search” → “For Files or Folders …” (Classic Start menu)
In the left hand panel of the “Search Results” window select “All files and folders”
In the “All or part of the file name” text box enter “.exe”
In the “Look in” pull down selection box select all hard drives on the system”
Select “When was it modified?”
Select “Specify dates”
In the pull down selection box Select “Created Date”
In the “from” text box enter the date the application installation started.
In the “to” text box enter the date the application installation was completed.
Select the “Search” button
(Repeat above steps looking for “.com” and “.dll” in place of “.exe”)

The requirement is **not** met if:

- Installed software is not approved under the DoDIIS Profile.

Or

- installed software matches the release and version of previously installed software; and
- Installs without prompting the user,

Or

- the installation process automatically installs additional COTS or GOTS software without checking if the software is already present.

INST-16 Installation of the application shall not replace shared resources. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION

An application shall not replace or modify a resource such that it is configured solely for the preferences of that application and no other. This reasoning is applied to resources such as utilities, environment declarations, and configuration files that may be used by more than one application. This includes not only the resources provided by the OS, but also the resources that are provided by the common infrastructure.

This requirement has broad uses. It applies to system-wide resources such as OS functions like printing command shells and X11 resources, and it also applies to resources that are tailored for each user such as .Xdefaults files and desktop preference settings..

TEST METHOD

Inspection of workstation resources will include files that are referenced during booting and initialization of the workstation. These files include inittab, ttytab, and inetd.conf, as well as resources that are referenced by OS services and user applications during startup and execution, including XKeysymDB, Xdefaults, and user preference files such as .cshrc. Appending application specific information to resource files is acceptable. Modifying objects that may be referenced by other applications is not acceptable.

The JAT results directory may be used to expedite the examination of application files.

UNIX:

```
   diff /usr/openwin/lib/X11/XKeysymDB ${X}/XKeysymDB | grep –v “!” | sort –u > /tmp/Xkdiffs
```

(where ${X} is the application directory containing the XKeysymDB file)

Microsoft:

The JAT results directory may be used to expedite the examination of application files.
Review the list of files that were modified during the application installation, and investigate those that may pertain to resource sharing.

In the registry, Examine the following key and subkeys:

HKEY_LOCAL_MACHINE\SOFTWARE\

Examine the various settings in Settings → Control Panel before and after application installation for modification
Folder Options
Display
System (e.g. Advanced Tab “User Profiles”)

INST-17 Not applicable for Version 3.0 and above test procedures. Added to Requirement INST-7.

INST-18 Not applicable for Version 3.0 and above test procedures.

INST-19*Application files shall be contained in a compliant directory structure. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-3

REQUIREMENT CLARIFICATION

On UNIX systems, the application directory structure will be compliant with the following format:

<root_dir>/application(or the application’s segments if DII COE segmented)
(where *root_dir* complies with the directory conventions defined by the infrastructure –
e.g. /h/GOTS or /h/COTS for AFDI-).

As a result, an application that is exported to client workstations shall be located in */export/<root_dir>/hostname#/ application_name*. The phrase “hostname#” simplifies distinguishing between network file (NFS) servers and between disks on the same server by using the disk number (e.g., */export/opt/main_server1/amhs*). These conventions clarify the administration of exported applications and simplify the use of the automount function provided by Unix OSs. This convention applies to all directories found under */opt*. For example, if application executables are located on a server, the executable path would be */export/opt/server_name/bin*, assuming that only one file system on the server is used for exported files.

There is no equivalent “hostname#” directory structure for an application that is exported to client workstations under Air Force DoDIIS Infrastructure (AFDI).

On Microsoft OSs, the application shall be contained in %systemdrive%\Program Files\application_name, where %systemdrive% is the drive identifier where Microsoft OS is installed.

Applications files should not be hard coded to a particular compliant directory structure. Rather, all defaults and installation instructions should point to a compliant directory structure for that OS, but remain changeable by the installer to permit sites flexibility to address mission specific requirements.

TEST METHOD

To verify the location of application files, execute the command:

UNIX:

```
# find / -name application_name (or segment name – repeat the find command for each segment)
   where "application_name" is the name of the base
   directory containing application files
    where "segment name" is the name of one of the application's DII COE segments
or

# cd  /<root_dir>/application_name  or
# cd /<root_dir>/hostname#/application_name
(where <root_dir> corresponds to the root directory defined by the infrastructure)
# ls –latR
```

Microsoft OSs:

Start→Find→Files or Folders …

Enter the application name in the 'Named' field and select the appropriate hard drive in the 'Look in' field. Verify that the base directory is located under %systemdrive%\Program Files.

INST-20*Not applicable for Version 4.2 and above test procedures.

INST-21 Not applicable for Version 4.0 and above test procedures.

INST-22$^{\#}$ The application shall not require specific settings of permissions and ownership of browser files and directories. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION

File and directory permissions and ownership must be set in accordance with the site security policy. Default directory permissions after a browser installation enable users to do things such as download plug-ins as needed. This may violate the site security policy, and permissions must be set to conform to the site security policy. The application design must take this and related file or directory configurations into account and be sufficiently robust in order to function properly with any adequate browser that has been installed and configured per site policy.

TEST METHOD

The permissions and ownerships of infrastructure files and directories will be recorded before the application is installed. Following successful installation of the application the infrastructure files and directories will again be examined to determine if any file or directory permissions or ownership has changed.

The following must be done on the base directory of infrastructure files. NOTE: it is recommended that the disk cache be cleared prior to the initial obtainment of file and directory permissions to reduce the number of files returned.

UNIX:

```
# cd [directory containing infrastructure files]
# ls –latR
```

Microsoft:

```
  cd [directory containing infrastructure files]
  > for /R %f in (*) do cacls %f
```

INST-23 Not applicable for Version 3.0 and above test procedures.

INST-24[#] Installation of the application client shall not overwrite or modify default browser configuration settings of any user. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION

Browser configuration settings are typically accomplished by each user rather than as global settings. The installation of the application client should not include an automated modification of any user’s default browser configuration settings including mail and news settings. Such changes may conflict with either the user’s preferences or with site policy. Instead, the application documentation should provide sufficient information that each user can set his/her browser preferences or settings appropriately.

TEST METHOD

Prior to installing the application, the user will view the browser ‘preferences’ and/or ‘Internet Options’ and perform the following steps to record the default settings. After the application has been installed and is ready for the general user, the user will repeat the aforementioned steps to verify that the default settings were unchanged.

All Operating Systems:
Verify that the time stamps on files in the user’s ‘Temporary Internet files folder’ were not changed during the installation. Special attention should be paid to the bookmarks.html, cookies, plugin-list, prefs.js, preferences.js and registry files, where applicable.

Example folder locations are

```
$HOME/.netscape/
c:\documents and setting\%USERNAME%
c:\program files\netscape\
```

The above procedure will be performed for each browser installed on the test workstation.

Microsoft:
In addition to the items above, verify default browser settings in the registry:

1) execute ‘regedit.exe to start the registry editor
2) within ‘regedit’, record the value of HKEY_CLASSES_ROOT\http\shell\open\command.

The value should be similar to:
E:\Program Files\Netscape\Communicator\Program\ netscape.exe –h “%1”

or

“E:\PROGRA~1\Plus!\MICROS~1\iexplore.exe” –nohome

3) within ‘regedit’, record the value of
HKEY_CLASSES_ROOT\http\shell\open\ddeexec\Application

The value should be similar to ‘NSShell’ or ‘Iexplorer’

4) within ‘regedit’, record the value of
Open HKEY_CLASSES_ROOT\http\DefaultIcon

The value should be similar to:
E:\Program Files\Netscape\Communicator\Program\ netscape.exe,0

or

%SystemRoot%\system32\url.dll,0

5) Repeat the above steps for HKEY_CLASSES_ROOT\https\...

If the application does not use a browser this requirement is Not Applicable.

INST-25 Not applicable for Version 4.0 and above test procedures. Incorporated into INST-24.

INST-26#*The web server directory structure shall be separate from the HTML documents directory. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION

The http configuration directory is typically separated from the HyperText Markup Language (HTML) documents directory in order to prevent web users from inspecting the server configuration files and discovering potential vulnerabilities.

TEST METHOD

Following installation of the application server, the HTTP configuration will be examined to determine that the HTML documents directory is separate from the HTTP server directory.

(Apache – UNIX)

There are 3 configuration files, (httpd.conf, srm.conf and access.conf), that can contain these server settings. The following commands will return the appropriate settings that should be compared:

```
# cd <HTTP server root directory>/conf/
```

(e.g. *HTTP server root directory* = /opt/WWW/apache)

```
# grep "^DocumentRoot" *.conf
# grep "^ServerRoot" *.conf
```

(Netscape servers)
<*server_root*>/admin-serv/config/ns-admin.conf
(e.g. *server_root* = /opt/suitespot)

(Microsoft IIS)
The IIS web server places HTML files in the C:\InetPub\Wwwroot directory by default.

This requirement is evaluated only on the server when a central server with browser access only configuration is used.
This requirement is Not Applicable if the application does not use a web server.

INST-27$^{\#}$* An “index.html” file or equivalent capability shall be used to control default web pages. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-3

REQUIREMENT CLARIFICATION

The use of a web interface to the application server should not permit a general user to browse through the server’s directories and files. The existence of an “index.html” or equivalent file in the directory eliminates the ability of a user to obtain listings of

directories and files on the web server. This file is specified in the server configuration. Without this file, if the URL for the web server specifies only a directory, then the httpd daemon returns a listing of that directory back to the user.
If a file other than “index.html” is used, then this file should be specified in the documentation provided by the application.

```
e.g.: …/apache/etc/srm.conf
        DirectoryIndex index.html index.cgi
```

In the case of Microsoft IIS, the default home page may be default.htm or default.asp instead of index.html.

TEST METHOD

Following the installation of the application server, the application documents directories will be examined to verify the existence of the “index.html” file in each directory under the Document Root directory.

If the index.html file is not present, then the ‘access.conf’, ‘httpd.conf’ and ‘srm.conf’ files in the server configuration directory will be examined to verify that an index file is specified. The application directories will be examined to verify that this file exists in each directory under the Document Root directory.

After the application server has been installed, the tester will attempt to browse the server directories by forming URLs from segments of the absolute path to web directories. The requirement is met if the tester is unable to obtain a listing of any directory accessed on the web server.

For centralized server with browser access only configuration this requirement is evaluated on the server.

This requirement is Not Applicable if the application does not use a web server.

INST-28$^{\#}$*All URLs referencing remote hosts shall contain the fully qualified domain names. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION

Depending upon its implementation/configuration, the browser may permit different settings for intranet (i.e., web sites within an organization’s network) versus internet (i.e., web sites outside an organization’s network). Settings for intranet web sites may be less restrictive than those for internet access (e.g., clients are allowed to execute Java applets from intranet sites but not from internet sites).

One method used by Internet Explorer to determine if the site was intranet or internet was by the presence of a ‘.’, if one did not exist, the site was considered to be intranet. A complete hostname in the URL will remove the ambiguity between intranet and internet access. The hostname must be specified as a logical host name rather than as a numeric Internet Protocol (IP) address. Use of logical host names makes underlying hardware and IP address changes transparent to the end users. Administrators simply update the site’s name resolution service to point the old name to the new IP address.

TEST METHOD

The application will be executed through the browser. A representative set of web pages will be traversed and each URL will be noted. The expansion of each URL will be examined to ensure that it identifies the domain name, and allows the viewer to determine whether the link points to an internet or intranet address.

For centralized server with browser access only configuration this requirement is evaluated on the server.

This requirement is Not Applicable if the application does not use a web server.

INST-29 Not applicable for Version 3.0 and above test procedures. Combined with ENV-5.

INST-30 Not applicable for Version 3.0 and above test procedures. Converted to INTSEC-16.

INST-31 Not applicable for Version 3.0 and above test procedures. Converted to INTSEC-17.

INST-32 Not applicable for Version 3.0 and above test procedures. Converted to INTSEC-18.

INST-33[#] Not applicable for Version 4.2 and above test procedures.

INST-34 Readme files and errata sheets shall contain only last minute and errata type information that could not be incorporated into the final printing of the official configuration and installation guide. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-3

REQUIREMENT CLARIFICATION
Readme files and errata sheets should not be used for whole portions of the configuration and installation document. Instead, these instructions should be in the formal configuration and installation guide. Typical use of readme files are for last minute and errata type information that could not be added to the deliverable guide before it was printed.
TEST METHOD
The contents of the readme files and errata sheets will be reviewed during the installation of the application. The requirement is met when the configuration and installation is successfully completed using the configuration and installation document with minimal information, or no information, taken from readme files and errata sheets.

INST-35$^{\#}$ The media delivered by the PMO to the JITF/NIMA ITF will contain only the complete baseline for the release version under test. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION
The PMO will deliver to the JITF/NIMA ITF media that reflects the delivery to user sites. The media will include all necessary software and data needed to complete the installation, and will not contain any superfluous information.
TEST METHOD
After installation of the application, the tester will determine if all data required for the installation was available. The media will be reviewed for superfluous information.

INST-36 The installation and configuration of the application shall be completed within the installation time estimate documented in the installation and configuration guide and must not exceed 20 working hours. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION
Installation and configuration covers the entire processing of loading software and modifying configuration files and parameters for successful operation of the application. It does not include loading of application data. The 20 hour limit is 20 sequential hours. If the installation is permitted to execute overnight (e.g., to extract software from media), the overnight hours are included in the time required to install the application. A realistic estimate of the time needed for installation and configuration of an application eases the burden of resource planning for system administrators.
TEST METHOD
The application installation and configuration guide will include an installation and configuration time estimate, not to exceed 20 hours. If no installation time estimate is given, this requirement is not met. The date and time at the beginning of the installation will be recorded. Once the application has been installed and configured, the date and time will again be recorded. Installation is completed after all required steps in the installation and configuration guide are performed successfully AND software verification is performed successfully. The time required to execute the software verification steps is not included in the time to install the application.

**INST-37 ***The application under evaluation shall not prohibit installation and operation of the application on a platform shared by other applications. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION
One goal of the common infrastructure is to give the sites flexibility in selecting how each application will be installed and used. An application that, by design, permits sharing of a platform with other application servers allows sites to select platforms based upon

application performance and resource usage. An application that, by design, requires a dedicated platform may hinder integration of the application into a site simply because computing resources – i.e., platforms and software – are duplicated unnecessarily. Resource sharing by applications should include more than simply coexisting on the same platform. It should include sharing computing resources such as data servers.

Additionally, dedicated servers should be avoided where mirroring of the application’s server is required to insure 24/7 availability and disaster recovery. Sites may not have the resources to dedicate a second server platform solely for the use of one application.

TEST METHOD

Application configuration and installation guide will be inspected to verify that the ability to share a server platform is specified. During installation and configuration of the application, the test engineers will note the configuration parameters that will prevent the application to operate on a platform shared with other applications.

INST-38# The application installation must result in a usable application. (UNIX and Microsoft OSs).

IMPACT CODE RANGE: 1

REQUIREMENT CLARIFICATION

The application installation instructions must be sufficiently detailed to allow for successful installation and operation of the application. It must be demonstrated that the installation was successful and that the application operates as expected. This is normally accomplished by executing a series of verification procedures that can be included in the installation documentation or provided as a separate document.

TEST METHOD

Upon completion of installation and configuration, the application will be started. Verification procedures will be executed and application operation will be observed.

INST-39#*There shall be no hard coded IP or URL addresses (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION

Use of logical host names and addresses makes underlying hardware and IP address changes transparent to the end users. Administrators simply update the site’s name resolution service to point the old name to the new IP address.

TEST METHOD

Application files will be searched for the presence of hard coded IP addresses. In the case of URLs, a representative set of web pages will be traversed and each URL will be noted. The expansion of each URL will be examined to ensure that it does not identify a hard coded address.

INST-40[#]*Application Databases shall Install or Update(Convert) correctly (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION

Application Databases shall install or update(convert) correctly.

New Installations:
Installation instructions or automated procedures shall correctly create the application’s databases, tables, rules and triggers within the COTS database server product. All data for the new databases shall load correctly from the installation media.

Updates(Conversions) -- 3 Possible Types: [Subsequent versions of an application, or patches to an application, often entail Updates(Conversions) to the Database]

1. Changes to the application’s databases, tables, rules or triggers – Instructions or automated procedures shall correctly modify the database. Data in the database shall be correctly shifted, deleted or otherwise modified to accomodate the change (e.g. Conversions of tables to include/remove/modify rows/columns between versions of the application – Table XYZ has two columns added, one column removed and one column’s data type changed from integer to real numbers)

2. Change to new version of the COTS database server product (e.g. Oracle 8i to Oracle 9i) – Instructions or automated procedures

shall properly convert/re-install the applications databases, tables, rules, triggers and data. 3. Conversions between COTS database server product vendors (e.g. Sybase to Oracle) – Instructions or automated procedures shall properly convert/re-install the applications databases, tables, rules, triggers and data.
TEST METHOD
Application databases shall be installed or updated(converted) in accordance with the instructions and automated procedures. The application's installation verification procedures will be run to determine if the database installed or updated (converted) correctly. This requirement is Not Applicable if the application does not Install or Update(Convert) a database

3.3 ENVIRONMENT

ENV-1*The application shall not modify system files in any way that causes the computing platform to fail to boot if the application client or application server is unavailable. (UNIX only)

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION

An application cannot assume that it “owns” the platform or platform resources. The workstation or server is a user tool, and accessing a specific application is only part of what a user may do during a login session. Since all applications at the site are integrated into the operating environment, the inaccessibility of a particular application does not mean that the user will not be able to perform useful work. The actual booting of the workstation must not be dependent upon the accessibility of any or all application servers. Likewise, a server platform may host one or more server application. Even on a server platform, the booting process must not be modified to halt or in some way hinder the boot process if the server application is unavailable for some reason.

TEST METHOD

The application configuration and installation guide will be reviewed to determine if any boot files are modified by the installation. The documentation will also be examined to determine what workstation resource files are modified by the installation. Following installation of the application, the boot files of the workstation will be examined to determine if the modifications made by the application installation process will prevent booting if the application server is unavailable. The files examined will include the init files for the OS:

UNIX: Execute the following commands to determine if any boot files have been modified:

```
sh
# for I  in /etc/rc* /sbin/rc* /etc/services /etc/*.conf
> do
> find $I \( -mtime –X –o –ctime –X \) –exec ls –latR { }“;”
> done
```

(where X represents time in days). Examine any files returned by the above commands.

After successful configuration and installation of the application, on both a server platform and on general user workstations, perform the following:

Halt a general user workstation. Halt the host on which the application server executes. After the server host has halted, reboot the user workstation. The workstation will complete its boot sequence and the login screen will be displayed.

This requirement is Not Applicable for Microsoft OSs

ENV-2# Execution of the application under evaluation shall not replace or alter system resources that are used by other applications. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION

An application shall not replace or modify a resource such that it is configured solely for the preferences of that application and no other.

This requirement applies to workstation resources such as utilities, environment declarations, and configuration files that may be used by more than one application. This includes not only the resources provided by the OS, but also the resources that are provided by the common infrastructure. OS and infrastructure patches are also covered by this requirement; the application cannot back out a patch and replace it with a newer version.

The requirement applies to system-wide resources such as OS functions like printing command shells and X11 resources and to resources that are tailored for each user such as .Xdefaults files.

TEST METHOD

UNIX:

The truss command, (e.g. truss –f –e –a –o *output file* [*application_name OR –p process_id])*will be used to identify files that are opened for writing by the application. For each file that is a system or user resource, the test engineer will verify that the application does not overwrite the file or replace any information in the file that is not specific to the application.

Microsoft :

The test engineer will ensure that the application(s) are started after the pre-run data is obtained and before the post-run data is obtained.
By comparing the pre/post results, the test engineer will verify that the application does not overwrite or replace any system resource.
The test engineer will verify that patches have not been backed out during the application installation.

ENV-3 The application shall not prevent or alter login if the application server or client is unavailable. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION

Stopping the execution of the application server software, halting the host on which the application server executes, or modifying the client application configuration so that the application client software is unavailable will not affect the user's ability to login to the workstation.

TEST METHOD

After successful configuration and installation of the application on both a server platform and on general user workstations, perform the following:

Stop the execution of the application server software. The OS and other services of the host on which the application server executes will still be available. After the application server has stopped, ping the host to verify that it is running and accessible. Login to a general user workstation. The login will complete normally and the user will be presented with the session environment and desktop, if one is configured for that session.

Halt the host on which the application server executes. After the host has halted login to a general user workstation. The login will complete normally, and the user is presented with the session environment and desktop if one is configured for that session.

Restart the server host and the application server software. On a general user workstation, modify the client application configuration so that the application client software is unavailable. This can be done by either a) moving the client executable file(s) to an inaccessible location on the user workstation or b) temporarily renaming the client executable file(s). If the client server is obtained via file sharing from an application server, either a) or b) must be done on the application server. Access to the application server is not altered. Once this has been completed, log out of the workstation. Login to the general user workstation as a general user. The login will complete normally, and the user is presented with the session environment and desktop, if one is configured for

that session.

ENV-4 The client application(s) of the application shall launch from the background menu or from an icon on the desktop. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2

REQUIREMENT CLARIFICATION
This requirement verifies that the client applications for the application will launch successfully from the background menu selection or by initializing the application from an icon on the desktop.
TEST METHOD
Following configuration and installation of the application on the general user workstation, the background menu item(s)/icon corresponding to the application will be selected. Selected test cases from the application test plan will be executed if normal operation of the application is not readily apparent. The requirement is not met if the application can only be started by the user from a command line.

ENV-5 Any application required daemons shall start automatically. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-3

REQUIREMENT CLARIFICATION
Daemons should start automatically in order to be available to requests from users at all times when the platform is operating. A daemon can be started at the time the platform boots (e.g., by execution of a boot script during system booting). It can also be spawned by a system process (e.g., “inet.d”) whenever a user request is received. The administrator should not be required to manually start the daemon for normal operation.
TEST METHOD
If the application design implements restart of the daemons or processes for the application during system reboot, the platform will

be halted and rebooted. Following the completion of the reboot, the process table will be examined.
If the application daemons or processes are spawned by a system process upon receipt of a user request, the platform will be set in an idle state (i.e., no user requests are being processed or are pending). The process table will be examined to verify that no daemons or processes for the application are executing. A request for data will be transmitted from a client application for the application. The process table for the platform will be examined again to verify that application daemons/processes are now running.

The requirement is not met if daemons or processes for the application must be started manually.
The requirement is met if the daemons or processes for the application are executing.

ENV-6*Application environment variables shall be defined at launch time and in the form of PRODUCT_VARNAME. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-3

REQUIREMENT CLARIFICATION
For UNIX systems, developers should assume that the following variables are global and have been defined by the site: PATH, HOME, TERM, TZ, LOGNAME, SHELL, and TMPDIR. The developer shall only define variables that are specific to the application and follow the format specified in this requirement. By following the variable naming convention, the probability that the application may overwrite or redefine variables of other applications is limited. Note that variables that are defined locally to the execution of the application (e.g., from a launch script) will not conflict with variables that are defined either globally or locally by other applications. Local definition of variables is preferred to globally defining variables that have meaning only to one application. For Microsoft OSs, there are several environment variables reserved: ComSpec, LOGONSERVER, HOME_DRIVE, HOME_PATH, NUMBER_OF_PROCESSORS, OS, PATH, PATHEXT, PROCESSOR_ARCHITECTURE, PROCESSOR_LEVEL, PROCESSOR_REVISION, SYSTEM_DRIVE, SYSTEM_ROOT, TEMP, TMP, USERDOMAIN, USERNAME, USERPROFILE, WINDIR

TEST METHOD

After completing the installation, environment variables defined by the application will be verified to be in the form of PRODUCT_VARNAME.

Following configuration and installation of the application, system environment variables and launch scripts used to invoke execution of the application will be examined to verify that all environment variables initialized follow the required format. The examination will include any data added to the infrastructure session management configuration files during the configuration and installation of the application.

UNIX:
Additionally, the truss command can be used to capture the environment settings. In order to follow an application’s activity, truss should be started in the following way:

```
truss –f –e –a –o output file [application_name OR –p process_id]
```

where

-f follows all child processes forked by the application
-e outputs the environment (i.e., the values of environment variables) of each forked process
-a outputs the arguments of each exec’ed process
-o gives the name of the file to which all output is written
-p identifies the process id of the process to be traced

The output of truss can be used to list the values of all environment variables by searching for “exec” calls. (In order to output the variables of the parent (initial) application, truss must be used to start the application, rather than simply attaching to a currently running process.)

Microsoft:
In addition to the above screening, the test engineer will perform the following (make sure all applications are closed):

1) Start → Run → cmd

2) In the command prompt enter:

```
>Regedit /e \[path]\[file].pre “HKEY_CURRENT_USER”
```

3) Start the application(s) and perform the following at the command prompt:

```
>Regedit /e \[path]\[file].post “HKEY_CURRENT_USER”
```

By comparing the files (/temp/pre_hkey_current_user.txt with /temp/post_hkey_current_user.txt), the test engineer will verify that the application does not overwrite or replace any reserved environmental variables.

ENV-7 Not applicable for Version 3.0 and above test procedures.

ENV-8*Not applicable for Version 4.2 and above test procedures.

3.4 OPERATION

OPS-1*Application file names shall consist of valid characters for file names and shall be restricted to the maximum length of 128 characters for UNIX/Solaris systems and 255 characters for Microsoft OSs. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION
This requirement is a general requirement for all files stored on a workstation or server. Valid characters for file names on UNIX/Solaris are defined in the X/OPEN XPG4 recommended character set, and in the Microsoft Logo specifications for Microsoft OSs. Valid characters are 0-9, Aa-Zz, . (dot) + (plus), - (minus), : (colon) and _ (underscore). Other characters are invalid because they may have meaning as meta characters, have meaning to the shell, or be difficult to reproduce (i.e., hidden characters). On Microsoft OSs, $ and space characters are acceptable.
TEST METHOD
Verify that the application under evaluation complies with requirements for filename structure and length. Navigate to the results directory created by JAT or by the manual steps run prior to application installation. Identify the differences between the pre_list.txt file and the post_list.txt file. (This will be a list of files that the application has added.) Verify that filenames do not exceed the 128 character limit. Verify that filenames use only valid characters.

OPS-2*The application shall use the platform's native keyboard map. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-3

REQUIREMENT CLARIFICATION

On Unix platforms, the keyboard, including the mouse buttons, is owned by the X server, but it is a shared resource. The list of key symbols (keysyms) associated with a specific keycode can be changed by any application. Since the keyboard is a shared resource, any changes made by one client application are global to all applications.

The default keysyms are defined in /usr/lib/X11/XKeysymDB (or /usr/openwin/lib/X11/XKeysymDB). Applications may append (but not overwrite) to this file, or may actually refer to a different XKeysymDB file, providing that this reference is not global to all applications. The most common change is to provide a more complete XKeysymDB file than the default. This will not constitute failure of this requirement. Most applications will have no need to use anything but the default XKeysymDB file. In any event, remapping of keyboard values should be done in such a manner that the changes are discarded upon application exit.

Under Microsoft OSs, there is no file map file. File map information is maintained in the Microsoft OSs registry. However, it is possible for an application to modify the native mapping of characters for the specific application.

TEST METHOD

Unix :

Typically, keyboard map modification is done in an application launch script via the “xmodmap” utility. To evaluate this requirement, execute the command:

```
    cd /<scripts directory>
    grep xmodmap *
```

If this command finds any xmodmap commands in the application’s scripts, the application is likely modifying the keyboard map. This can be determined by the options passed to the xmodmap command. The –e option is used to change either a keysym listing or a mapping of keysyms to a keycode.

Alternatively, the xmodmap command can be used to capture the current keyboard map. Prior to starting the application, execute the following commands:

```
    xmodmap –pm >/tmp/mod.map  (modifier map)
    xmodmap –pk >/tmp/key.map    (keyboard map)
    xmodmap –pp >/tmp/pointer.map  (pointer or mouse map)
```

After starting the application, repeat the three commands in a separate command window and save the output to three different files (e.g., mod1.map, key1.map, pointer1.map). Compare the contents of the pairs of maps by either inspection or via the “diff” command. If the application has not changed any of the maps, then there will be no differences.

The application may append keysym entries to the default XKeysymDB file. Compare the XKeysymDB file prior to application installation to the file after the application has been installed. The requirement is not met if any keysym entries have been overwritten.

The application may install and use a different XKeysymDB file than the one found in ***/usr/lib/X11***. The application must set the environment variable XKEYSYMDB to the path of this alternate file.

This variable must be set locally; the requirement is not met if the variable is set globally. The variable is set globally if it is initialized at the time of user login. To determine if the variable has been set globally do the following:

On the command line before starting the installation enter:

```
    echo $XKEYSYMDB
```

Verify that the variable has no value.

Microsoft:

The registry will have been exported to ‘pre_reg.txt’ before the installation and then to ‘post_reg.txt’ after the installation. Values in these two files will be compared to evaluate this requirement.

For the mouse(NT):

Using the “find” feature of MS Wordpad, locate HK_LOCAL_MACHINE\HHARDWARE\DeviceMap\PointerPort in ‘pre_reg.txt’ Record the data path to all the values listed (i.e. \REGISTRY\Machine\System\ControlSet001\Services\i8042prt).

Using “find” again to locate each, record the following value/data pairs of the Parameters key for each entry recorded above (i.e. \REGISTRY\Machine\System\ControlSet001\Services\i8042prt\Parameters)

MouseDataQueueSize	(100)
NumberOfButtons	(2)
PointerDeviceBaseName	“PointerPort”

SampleRate	(40)
MouseResolution	# if present

Locate and record all the value/data pairs listed in HK_CURRENT_USER\Control Panel\Mouse

Compare these values to the corresponding values in ‘post_reg.txt’

For the Keyboard(NT):

Using the “find” feature of MS Wordpad, locate HK_LOCAL_MACHINE\HHARDWARE\DeviceMap\KeyboardPort in ‘pre_reg.txt’. Record the data path to all the values listed (i.e. \REGISTRY\Machine\System\ControlSet001\Services\i8042prt)

Record the following value/data pairs of the Parameters key for each entry recorded above
(i.e. \REGISTRY\Machine\System\ControlSet001\Services\i8042prt\Parameters)

KeyboardDataQueueSize	(100)
OverrideKeyboardType	# If present
OverrideKeyboardSubtype	# If present
KeyboardDeviceBaseName	“KeyboardPort”

Locate and record all the value/data pairs listed in HK_CURRENT_USER\Control Panel\Keyboard
Compare these values to the corresponding values in ‘post_reg.txt’

For the mouse(Windows 2000/XP):

Use regedit to find the value of HKEY_LOCAL_MACHINE\SYTEM\Select\Current. This value will establish which Control Set is being used and/or modified currently (i.e. a value of “1” points to ControlSet001)

Using the “find” feature of MS Wordpad, locate “MouseDataQueueSize” in ‘pre_reg.txt’. Record the following value/data pairs of the Parameters key for ControlSet00x, where “x” is the value of “Current” found above.

EnableWheelDetection	(2)
FirmwareIdentified	(1)
Migrated	(1)
MouseDataQueueSize	(100)
MouseInitializedPolled	(0)
MouseResolution	# if present
MouseSynchIn100ns	(20000000)
SampleRate	(40)
WheelDetectionTimeout	(1500)

Compare these values to the corresponding values in ‘post_reg.txt’

For the Keyboard(Windows 2000/XP):

Using the “find” feature of MS Wordpad, locate “MouseDataQueueSize” in ‘pre_reg.txt’. Record the following value/data pairs of the Parameters key for ControlSet00x, where “x” is the value of “Current” found above.

FirmwareIdentified	(1)
KeyboardDataQueueSize	(100)
Migrated	(1)
PollStatusIterations	(1)

Compare these values to the corresponding values in ‘post_reg.txt’

OPS-3[#] The execution environment that exists at the time of application launch shall not conflict with either the user’s overall operating environment or the execution environment of other applications. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION

The execution environment of the application is defined by the environment variables set by the OS, the infrastructure, and the application. The execution environment should not result in ambiguous or incorrect references to commands or files due to assumptions by the application with regard to environment settings.

TEST METHOD

Review the launch scripts for definition of global variables and reference/modification of shared resource files.
Evaluate changes (if any) in the application’s processing parameters.
Evaluate the integration of the application into the infrastructure sessions and the associated definition of global variables.

UNIX:
Execute ‘set’ and at a minimum note the following variables: PATH and LD_LIBRARY_PATH, or run the truss command to capture environment variables: truss –f –e –a –o *output file* [*application_name OR –p process_id]*
where
-f follows all child processes forked by the application
-e outputs the environment (i.e., the values of environment variables) of each forked process
-a outputs the arguments of each exec’ed process
-o gives the name of the file to which all output is written
-p identifies the process id of the process to be traced

The output of truss can be used to list the values of all environment variables by searching for “exec” calls. (In order to output the variables of the parent (initial) application, truss must be used to start the application, rather than simply attaching to a currently running process.)

Microsoft:

NT
Right-click on ‘My Computer’→’Environment’ tab, and record the system variables before and after the application is launched.

2000/XP**:**
Right-click on ‘My Computer’→’Properties’→’Advanced’(tab)→’Environment’ (bottom of window), and record the system variables before and after the application is launched.

OPS-4 The application shall not contain configuration files or tables that duplicate information already contained in the OS configuration files. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION

The application design should not include duplicate information that is already contained in and distributed by the common infrastructure. This includes information that is available from an OS service such as NIS/NIS+ and information that is maintained by other infrastructure services such as Domain Name Service. Duplication of this type increases the risk of losing synchronization with other applications that are utilizing the same information. For example, placing the name and IP address of the application server in an application configuration file can affect the execution of the application. An update to the application configuration file would also be required if the IP address is changed by the system administrator. Unless the application administrator has kept detailed configuration records, he/she may not be aware that this must be done until the application fails to execute properly.

TEST METHOD

The application design documentation and configuration and installation guide will be inspected to determine if any redundant information is being maintained by the application.

After the application has been installed, the configuration files created or modified by the application will be inspected for inclusion of redundant information. Redundant information will include, for example, host name/IP address pairs, reserved port numbers (except for the application itself), and the local host name.

OPS-5*The application shall not use extensions to the Window System that are not supported by the infrastructure. (UNIX only)

IMPACT CODE RANGE: 2-3

REQUIREMENT CLARIFICATION

X Window System extensions improve the ability of the workstation to display graphics such as postscript or animation. In order for applications to operate on any platform that uses the X Window System, the application must implement and comply with a common set of extensions.

The X Consortium defines a set of extensions to the X Window System. In order for an application to use any extension in this set, the X server must support the extension, and the necessary library must be present on the platform that is executing the application. The X server provided by the Solaris OS supports the following X extensions:

- Display Post Script (DPS)
- X Input Extension
- Double Buffer Extension
- Shape Extension
- Shared Memory Extension
- Miscellaneous Extension
- XC-MISC
- X Imaging Extension

The extensions require the libraries “libXext”, “libXi”, and “libdps*” in /usr/lib/X11 (/usr/openwin/lib/X11). These libraries are part of the infrastructure, and the application does not need to add them during installation.

TEST METHOD

If the application uses extensions to the window system that are not supported by the infrastructure X server, it must either place additional libraries in the standard system directories, such as **/usr/openwin/lib** or modify the library search path via the environment variable LD_LIBRARY_PATH. In addition, the X server must be modified or replaced to support the additional extensions.

After installation of the application, the directories that are touched during application configuration and installation will be examined to verify that the application does not include or bundle additional libraries for the window system extensions. The installation must not overwrite any OS libraries.

The native X server will be checked to verify that it has not been replaced during installation of the application. If the application installation includes loading of an X server, the documentation will be examined to determine if the execution of the application requires using this X server in place of the native X server.

The requirement is not met if the application adds additional X extension libraries to the platform during installation, overwrites the native X extension libraries, or if an additional X server is loaded on the platform during application installation and is required for execution of the application.

JAT results, if available will be used to expedite application examination.
This requirement is Not Applicable for Microsoft OSs.

**OPS-6 ***The application shall use the infrastructure print utility for printing hard copy. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-3

REQUIREMENT CLARIFICATION

This requirement is applicable for both application client and application server processes and assumes that destination printers are managed by the infrastructure print management utility. An application should not control or otherwise direct printing; this should be done instead by the infrastructure printing service.

TEST METHOD

Generate and inspect hard copy printouts for correct banner markings.

UNIX:

Compare the following files before and after running the application.

```
find . –name ‘lp?’ –a  –exec ls –ld {} ‘;’ > <pre-run file>
find . –name ‘lp?’ –a  –exec ls –ld {} ‘;’ > <post-run file>
diff <pre-run file> <post-run file>  >  <diff file>
more <diff file>
```

Microsoft:

Compare the following files before and after running the application.

```
        Example:  dir /t:w %systemroot%\system32\ winspool.drv > <pre-run.txt>
                  dir /t:w %systemroot%\system32\ winspool.drv > <post-run.txt>
                  fc <pre-run.txt> <post-run.txt>  >  < diff file.txt>
```

%systemroot%\system\winspool.drv
%systemroot%\system32\winspool.drv
%systemroot%\system32\spoolss.dll

NT only:

%systemroot%\system32\spool\prtprocs\w32x86\winprint.dll
%systemroot%\system32\spoolss.exe

<u>2000 only:</u>
%systemroot%\system32\spool\prtprocs\w32x86\sfmpsprt.dll

If there are no differences between the files before and after running the application, then the printing infrastructure is not being altered. The application's print functionality of can be compared to other installed applications (e.g. Microsoft Word).

OPS-7 Administration of the application shall not require access to superuser accounts. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION

Once the application is installed and configured, administrative functions that specifically address access to and operation of the application should not require logging in as root or as an administrator. This approach reduces the probability that administrative changes for one application may affect the operation of other applications or the operation of the workstation or server platform itself.

Access to application administration functions can be implemented in one of several ways:

1. A functional user ID can be used. This ID is placed in a restricted UNIX group for application administrative functions. In this approach, the administration functions are typically available through menu selections in an application window.
2. The user ID that is used for application administration is a separate user ID that reflects the greater privilege and trust required for application administration.
3. The application administration functions are accessible by user Ids that are associated with administration of site software. The use of an infrastructure trusted role is appropriate in this approach.

The application design may require a combination of the approaches listed above. For example, an application may provide administrative functions from its main window to certain user Ids and also require access to a privilege user ID for database administration.

TEST METHOD

After the application has been installed, executable files that provide administrative functions will be identified. The permissions on each file will be examined to verify that the application administrator does not require superuser (root on Unix and administrator on Microsoft OSs) privileges to manage the application.

UNIX:

```
   # ls –al    ;-verify permissions
```

Microsoft:

```
   c:\ cacls [filename(s)]    ;-verify permissions
```

OPS-8[#] The administrator shall be provided with utilities and tools to add, modify, or delete application users. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION
This requirement refers only to managing users of the application, not to the definition and management of workstation users (i.e., Unix or Microsoft OSs accounts). The latter is performed via the infrastructure user management tools. Many applications will not provide or need any tools other than infrastructure User Management. User management should be limited to doing what is needed to give the user access (or take away access) to the application and its data. If access can be achieved by using the already existing tools of the infrastructure, then no additional utilities are required. In the case of applications that rely on databases, the management tools of the database management application are sufficient, and the application does not have to provide additional, redundant tools.
TEST METHOD
The application administration documentation will be reviewed to identify the approach to application user management. The tools to add, modify, or delete application users will be identified. After the application has been installed, the identified tools will be located. The tools will be evaluated to determine if any of the tools is a redundant implementation of an operating service or infrastructure, including database management, service, etc. This requirement is Not Applicable if the application does not provide and does not require additional tools to manage application users.

OPS-9 The application shall use infrastructure management utilities to manage and distribute application, user, and security data. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION

The application developer must use the management services of the common infrastructure wherever it is appropriate. Since the trend is toward shrink-wrapped applications, there should be, in general, few requirements for an application to manage system resources such as user data and security data. Management requirements for the application must pertain solely to areas of management that are specific to the application rather than to areas of management that pertain to the system in general.

TEST METHOD

The appropriate application documentation (e.g., Software Design Document [SDD], Trusted Facilities User's Guide [TFUG]) will be examined to verify that application, user, and security management are performed with infrastructure management utilities. The administration tools provided by the application will be identified.

After the application has been installed, the administration tools will be exercised to evaluate their functions. Executing the tools will verify that the application utilities do not duplicate infrastructure tools to manage and distribute application, user, and security data.

OPS-10 Application execution shall not fill or result in exhausted file system space. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION

Many applications use files that are continually increasing in size. Such files are log files, temporary files, and audit files. If the application relies on the syslog file, temporary directory, and audit directories provided by the infrastructure, then managing these growing files becomes the system administrator's responsibility and is no longer the responsibility of the application. However, if the application places its logs, temporary files, and/or audit data in other locations, then the application documentation should clearly identify these locations. Additionally, the application design should account for these growing files and provide the means to automatically reduce them as needed.

DBMS transaction logs are also covered by this requirement. If the application implements a transaction log within the DBMS, then the application administration documentation must provide guidelines to ensure that the log does not exhaust space within the DBMS and stop the DBMS. This is particularly critical if the application is one of several applications sharing a data server; the transaction log associated with the application could crash the data server, thus causing disruption of service to other applications.

TEST METHOD

UNIX:

During execution of the application, the application process will be monitored via the “truss” process.
In order to follow an application’s activity, truss should be started in the following way:

```
truss –f –e –a –o output file [application_name OR –p process_id]
```

where

-f follows all child processes forked by the application
-e outputs the environment (i.e., the values of environment variables) of each forked process
-a outputs the arguments of each exec’ed process
-o gives the name of the file to which all output is written
-p identifies the process id of the process to be traced

To find rapidly growing files, the output would be searched for “write” calls. The test engineer will verify that each indicated file is managed to avoid exhausting file system space (e.g., deletion or compression of the temporary files).

If the application uses a DBMS, then the application administrator must be aware that the transaction logs must be managed.

The application administration documentation will be examined to verify that guidance for managing the transaction log is provided.

Microsoft:

Event Viewer logs automatically stop logging or purge themselves when the maximum log size value is reached. For applications that do not register their logs with the Event Viewer, review changes in the ‘difference file’ after exercising the application to generate logging. Execute the following commands:

```
dir /s /t:w /a > <pre-run file.txt>
```

Exercise the application to generate logging, then run:
dir /s /t:w /a > <*post-run file.txt*>
fc < *pre-run file.txt*> < *post-run file.txt*> > <*difference file*>*.txt*
Check the contents of < *difference file.txt*> for files that are used for logging. (Disregard files used by Event Viewer.)

If files exist that have the potential to exhaust file system space, verify that the documentation identifies the files to be managed.

OPS-11 The loss of connectivity between the application client process and the application server process shall not affect the behavior or operation of other client workstation applications or utilities. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION

Connectivity refers to the ability to pass protocol data units (e.g., packets, TCP/IP transmission units) between the application client process on the user's workstation and the application server process executing on either the same workstation or on another platform. From the perspective of the user, connectivity can be lost if the server process is terminated unexpectedly or if the network path between the two processes is broken in some way. The loss of connectivity should not cause other processes on the workstation, including the OS, to operate incorrectly, such as hanging or terminating unexpectedly. The application itself may hang or terminate depending upon the application design. For browser-based applications, the browser itself may hang. It is acceptable that the web access/transfer can be stopped or the window closed. In some cases, the browser may have to be terminated; this is outside the scope of this requirement.

TEST METHOD

The objective will be verified in two ways:

1. The application server process will be terminated during an application client session with the server without normal notification to the client. The operation of the user's workstation will be evaluated to determine that no process, other than the application client process itself are affected.
2. The network connection between the application server process and the application client process will be broken during a client session. This can be efficiently accomplished by disabling the network interface of the platform on which the server process is executing. This does not affect the operation of the network itself. The operation of the user's workstation will be evaluated to determine that no process other than the application client process itself is affected.

UNIX:

```
# ifconfig –a
```

Get the interface which contains the IP address of the host. (e.g. le0)

```
# ifconfig [interface] down (e.g. ifconfig le0 down)
```

Perform tests.

```
# ifconfig [interface] up  (e.g. Ifconfig le0 up)
```

Microsoft:

Remove the NIC category five cable to facilitate a loss of network connection.

Perform tests.

OPS-12 Disorderly termination of the application shall not affect the execution or behavior of other applications. (UNIX only)

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION

The activity of the application should not affect the activity of other applications executing on the same platform or in the same operating environment (i.e., the user site).

Disorderly termination can occur if the application exits due to a software error or invalid user action or if the application is unexpectedly halted by a user or administrator action. Other applications should continue to operate normally when such events occur.

TEST METHOD

This requirement will be verified in the following manner:

1. The application will be started in a typical user session. At various points in the session (e.g., initial startup, data transfer/review, query/response), the client application will be terminated by using the “kill” command from a shell window. For web-based applications, the browser is considered the client application.
2. The application will be started in a typical user session. At various points in the session (e.g., initial startup, data transfer/review, query/response), the user will log out of the workstation without first exiting the application.

In both cases, the operation of the user’s workstation will be evaluated to determine that no other processes are affected.

In order to test the effect of disorderly termination of the application server processes, the following steps should be followed for servers that are using the DBMS.

```
# cd …/sybase/bin/isql –Usa –P<sa password>
1>shutdown SYB_BACKUP          (To shutdown the backup server)
2> go
1> shutdown               (Shuts down the main data server)
2> go
# sync
# sync
# halt
```

If the data server is shared among several applications, then these applications will be affected by these steps.

Verify that applications and OS services running on the same platform as the data server are still running properly.

Restart the data server. Terminate the application server processes. Verify that the applications and OS services running on the same platform as the data server are still running properly.

This requirement is Not Applicable for Microsoft OSs.

OPS-13 Not applicable for Version 4.0 and above test procedures.

OPS-14 Orderly termination of the application shall not affect the execution or behavior of other applications. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION
If the normal process of starting and stopping the application affects the operation of other processes on the workstation or of the application itself when it is invoked again, the application design is unsatisfactory.

Sample test scenarios will be performed in which the application is started, used in typical manner, and then terminated by the recommended steps.
TEST METHOD
This requirement will be verified in the following manner: The application will be started in a typical user session. At various points in the session (e.g., initial startup, data transfer, query/response), the client application will be terminated by using the “exit” command or button from the application main window. The application server application will be started. While users are accessing the server via client application applications, the server will be shut down using the application’s documented steps for stopping the server. Following each scenario, the operation of the user’s workstation will be evaluated to determine that no other processes are affected.

OPS-15 Disorderly shutdown of the client workstation while the application is executing shall not affect the behavior or operation of the application on other workstations. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION
There should be no effects that are attributable to the application on other workstations if the user’s workstation is shut down while the application is active. Once the workstation or server platform is rebooted and the application is restarted, the application should execute normally.
TEST METHOD
The application will be started on the user’s workstation. Once the application is active, the workstation will be shut down (i.e., halted). The application processes on other workstations in the test environment will be evaluated for normal operation.

OPS-16 Disorderly shutdown of the client workstation while the application is executing shall not result in incorrect behavior of the application when the application is restarted. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION

There should be no effect on other workstations that are attributable to the application if the user’s workstation is shut down while the application is active. Once the workstation or server platform is rebooted and the application is restarted, the application should execute normally.

TEST METHOD

The application will be started on the user’s workstation. Once the application is active, the workstation will be shut down (i.e., halted). After the workstation is rebooted, the application is restarted, and the normal operation of the application will be verified.

UNIX:

```
# sync;sync;halt
```

Microsoft:

Power off and reboot

OPS-17$^{\#}$ User logout of the client workstation while the application is executing shall not affect the behavior of the application or the behavior of other applications in the user’s next login session. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION

Once the user logs in to the workstation and invokes the application, the application should execute normally. The application may not execute normally if the user logs out and consequent termination of the application leaves a residue of lock files and similar objects that will affect the behavior of the application. However, the application should be able to recover either by specific actions of the user or after a period of time. There should be no effect on other applications that are started in the user’s next login session

TEST METHOD

Test scenarios will be run in which the application is started and the user logs out at various points in the scenario. After the user logs back into the workstation, selected applications will be run, and their normal operation will be verified. The next scenario will be started by launching the application, and the normal operation of the application will be verified. Following the verification, the user will log out of the workstation at a different point in the scenario.

The requirement is met if, for all scenarios,:

(a) Normal operations of other applications are not affected, AND (b) Normal operation of the application is not affected. If the application does not operate normally immediately but does recover either by a user action or after a period of time, condition (b) is met. The requirement is not met if any processes associated with the application remain active after the user has logged out.

OPS-18$^{\#}$ * The application shall exhibit consistent behavior across all supported OSs and platforms. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION
The application design should enforce a uniform look and feel across all of the platforms and OSs supported by the application. Limitations due to the hardware and OS that prevent a uniform look and feel should be identified in the application design documentation. There should be no differences in the functions provided by the application to the user regardless of the platform and OS.
TEST METHOD
Ad hoc testing will be performed on each platform in the test environment that is supported by the application. A combination of testing and inspection will be used to verify that there are no differences in the application function regardless of the platform and OS.

OPS-19$^{\#}$ The application shall not duplicate functions provided by support applications. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION
A primary objective of establishing a common infrastructure and common support applications for user sites is to eliminate the redundant implementations of functions by applications. An application must only implement functions that are specific to its scope. Otherwise, it must use the services provided by the infrastructure support applications.
TEST METHOD
The application configuration and installation guide will be examined to verify that the application does not include functions that are

provided by support services, such as word processors, spread sheets, browsers, and file transfer utilities. After installation of the application, the application directories will be examined for modules that duplicate support services.

Verify that the application is not duplicating functions provided by support applications.
The results files generated by either JAT or the documented manual procedures will be used to expedite the application examination.
Examine the application directory tree and execute the command:

UNIX: find / -ctime -*n* | xargs file | grep "executable SPARC" | cut -f1 -d: | xargs ls –acld (Where *n* is the number of days ago the application installation was started.)

Microsoft:

NT:

Select "Start" → "Find" → "For Files or Folders …"
In the "Name & Location" tab:
In the "Named" text box enter ".exe"
I n the "Look in" pull down selection box select all hard drives on the system"
In the "Date Modified" tab:
Select the "Find all files created or modified:" radio button
Select the "between" radio button
In the "between" text box enter the date the application installation started.
In the "and" text box enter the date the application installation was completed.
Select the "Find Now" button
(Repeat above steps looking for ".com" and ".dll" in place of ".exe")

2000:

Select "Start" → "Search" → "For Files or Folders …"
In the "Search for files or folders named:" text box enter ".exe"
In the "Look in" pull down selection box select all hard drives on the system"

Select “Search Options >>”
Select “Date” check box
Select “files Created” from the pull down text box
Select the “between” radio button
In the “between” text box enter the date the application installation started.
In the “and” text box enter the date the application installation was completed.
Select the “Search Now” button
(Repeat above steps looking for “.com” and “.dll” in place of “.exe”)

XP:

Select “Start” → “Search”
or
Select “Start” → “Search” → “For Files or Folders …” (Classic Start menu)
In the left hand panel of the “Search Results” window select “All files and folders”
In the “All or part of the file name” text box enter “.exe”
In the “Look in” pull down selection box select all hard drives on the system”
Select “When was it modified?”
Select “Specify dates”
In the pull down selection box Select “Created Date”
In the “from” text box enter the date the application installation started.
In the “to” text box enter the date the application installation was completed.
Select the “Search” button
(Repeat above steps looking for “.com” and “.dll” in place of “.exe”)

Examine appropriate directories to determine if duplicate support services are being used.

OPS-20*The application shall use shared libraries for UNIX/Solaris and DLLs for Microsoft OSs. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 3-4

REQUIREMENT CLARIFICATION

Use of shared libraries, if supported by the OS, results in less disk space required to store the application.

TEST METHOD

Determine if shared libraries are used by application software. Following installation of the application, the application binary files will be examined using the “file” utility to determine if dynamic linking of libraries is employed.

UNIX:
To verify which application binaries use shared libraries execute the command:
file <binary name>

If libraries are dynamically linked execute the command:
(SOLARIS) ldd <binary name>
to determine which libraries are linked to the application.

Microsoft:
NT
Information can be derived from MyComputer\HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\SharedDLLs Registry Entry.

After installation of the application, the application directories will be examined for executable files. Identify the application executables.
“Start” → “Find” → “For Files or Folders …”
In the “Name & Location” tab:
In the “Named” text box enter “.exe”
In the “Look in” pull down selection box and use “browse” to select the application directory.
Select the “Find Now” button

In order to verify the application utilizes shared DLLs, the engineers will run a ‘Dependency Walker’ program such as ‘depends.exe’ in conjunction with every executable file found. (depends.exe can be found on the Windows NT 4.0 Service Pack 4 Note:

“depends.exe” is not part of the later Service Packs. Thus if the current service pack is loaded, instead of upgrading from the earlier Service Pack 4, “depends.exe” is not present).

2000:
Information can be derived from HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\SharedDLLs Registry Entry.

After installation of the application, the application directories will be examined for executable files. Identify the application executables:
“Start” → “Search” → “For Files or Folders …”
In the “Search for files or folders named:” text box enter “.exe”
In the “Look in” pull down selection use “browse” to select the application directory.
Select the “Search Now” button

In order to verify the application utilizes shared DLLs, the engineers will run a ‘Dependency Walker’ program such as ‘depends.exe’ in conjunction with every executable file found. ‘depends.exe’ is not supplied with Windows 2000 and must be run from the JAT CD or obtained from a 3rd party source.

XP:
Select “Start” → “Search” → “For Files or Folders …” (Classic Start menu)
In the left hand panel of the “Search Results” window select “All files and folders”
In the “All or part of the file name” text box enter “.exe”
In the “Look in” pull down selection box select all hard drives on the system”
Select the “Search” button

In order to verify the application utilizes shared DLLs, the engineers will run a ‘Dependency Walker’ program such as ‘depends.exe’ in conjunction with every executable file found. ‘depends.exe’ is not supplied with Windows XP and must be run from the JAT CD or obtained from a 3rd party source.

Note that if an executable does not reference a DLL, it does not mean the application failed the requirement. It is necessary to

consider what the function of the application is and if it is possible to utilize a DLL. Applications that have installable options typically store the code for the option in a DLL

OPS-21[#] The application shall not require use of a browser with acceptance of cookies enabled. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION
Many browser-based applications rely on cookies written by the web server and stored locally by the browser. This practice has been widely accepted and, at the current time, no security vulnerabilities relating to the use of cookies have been identified. However, site security policy may require acceptance of cookies to be disabled, and the application must be able to function properly with this restriction.
TEST METHOD
The browser will be configured to refuse cookies. Netscape: On the browser menu bar: Select *Edit* Using the pull down menu select *Preferences* Click *Advanced* to display the Cookie Options box Select the *Disable Cookies* option Click on the *OK* button. Internet Explorer(Version 6): On the browser menu bar: Select *Tools* Using the pull down menu select *Internet Options* Click the *Privacy* tab Move the slider to the top of the vertical bar Click on the *OK* button.

Internet Explorer(Other versions):
On the browser menu bar:
Select *Tools*
Using the pull down menu select *Internet Options*
Click the *Security* tab
Select the *Custom Levels* button
Scroll to the *Cookies* section of the list and click on the *Disable* option.
Click on the *OK* button.

NOTE: Cookies are stored in Netscape cache files on the UNIX version of Netscape; the PC version maintains a separate cookie file.

The application will then be accessed. The behavior of the application will be evaluated to verify that it is functioning normally.

This requirement is Not Applicable if the application does not use a browser.

OPS-22$^{\#}$ Web pages shall not contain animations and animated Graphics Interchange Format (GIF) files that do not implement mission functions. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION
System resources that are required to display animation may cause additional delays in downloading the objects that implement animation or may cause performance problems for the application or for other applications. Animations must be limited to those that are clearly necessary to accomplish one or more mission functions.
TEST METHOD
The execution of the application will be inspected to verify that animations and animated GIF files have functions pertinent to the scope of the application. This requirement is Not Applicable if the application does not have a web-based component

OPS-23[#] Web pages shall not contain elements that obscure or interfere with reading clarity. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION
This requirement emphasizes that application web pages should focus on mission functions rather than artistic additions that may distract from the application mission.
TEST METHOD
The execution of the application will be inspected to verify that application web pages do not contain over busy background patterns, low contrast between foreground and background, non-functional blinking text, or other elements that would impact reading clarity. Blinking text may be used to implement or enhance mission functions (e.g., a flashing security alert). This requirement is Not Applicable if the application does not have a web-based component.

OPS-24[#] Large graphic images shall be downloaded on demand. A small icon of the image shall be displayed on the web page and linked directly to the full-sized image. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION
Large graphic images may cause performance problems on resource-limited workstations or on bandwidth-limited network links. Providing links to such images allows the user to select which larger images he or she wishes to see. The image size of 50 Kbytes should be used as guidance for determining which images should not be downloaded automatically.
TEST METHOD
The execution of the application will be inspected to verify that large graphic images are not automatically downloaded to application web clients. Images larger than 50 Kbytes should not automatically download. If the application does not use a browser this requirement is Not Applicable.

OPS-25 Not applicable for Version 3.0 and above test procedures.

OPS-26[#] The application software and documentation shall explicitly identify the software version and release of the application. (UNIX and Microsoft OSs). NOTE: Converted from INST-8.

IMPACT CODE RANGE: 3-4

REQUIREMENT CLARIFICATION
A user site must be able to identify what it is installing and configuring to ensure that the software is current. This information ensures that the documentation and software are for the same version and release. This information is also necessary when reporting errors or problems to a software support facility or help desk.
TEST METHOD
This requirement will be evaluated by inspection of the software and documentation for version and release numbers. The information from both sources must match. Software items to examine include Splash Screens, About dialog box, and Help.

3.5 USER INTERFACE

GUI-1 Not applicable for Version 4.2 and above test procedures.

GUI-2 Not applicable for Version 4.2 and above test procedures.

GUI-3 The application shall display appropriate error messages when requested colors are not available. (UNIX only)

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION
The X server returns an error to an application when a request for a color cannot be serviced because no read-only or free color cells are available. The application can either terminate or display the built-in black and white colors. If the application terminates, then the correct reason for termination (i.e., colors could not be obtained) must be displayed. The error message can be displayed in the console window or in a popup window if possible. Applications should also write an appropriate message to the application audit trail.
TEST METHOD
The default color map will be filled with a sufficient number of read-write color cells so that the application is unable to obtain all of its requested colors. This can be done using either a test driver that allocates read-write cells or by starting several invocations of an application that is known to use read-write cells. Once the color map is filled, the application is started. The display of a suitable error message that describes the reason (i.e., cannot allocate colors) for termination will be observed. If the application sends audits via the infrastructure audit Application Program Interface (API), the audit file will be examined for accompanying audit messages reporting the termination of the application and the reason for termination. This requirement is Not Applicable for Microsoft OSs. This requirement is Not Applicable for web applications.

GUI-4[#] Application windows shall provide panning or scrolling methods to view panes larger than the available frame. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION
The application design should take into consideration the amount and dimensions of the information that will be displayed in application windows. Scrolling or panning methods should be provided for windows in which information output may either be too large to display completely or may scroll past before the user can read the window contents. Allowing the user to resize the window to display the full contents is an unsatisfactory solution, since there may be times when the largest window size is insufficient to display all of the output. Also, scroll bars are an indication that there is more output; it is possible that a user may not recognize that a window should be resized to view the complete output. Conversely, the application design should not place scroll bars on windows when the scroll bars would serve no purpose.
TEST METHOD
The application will be exercised to examine application windows in which information output is displayed. The presence or absence of scrolling or panning methods will be observed and the suitability or need for scrolling or panning methods will be evaluated.

GUI-5[#] The application shall support copy and paste between windows. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION
Meeting this requirement provides a user the ability to reduce errors resulting from incorrect data entries. In addition, the ability to copy and paste between windows will expedite data transfer between windows.
TEST METHOD
The application will be examined to determine if user is able to copy and paste between windows. **UNIX**: Highlight to copy, middle mouse button to paste

or use the copy/paste keys relevant to the platform. **Microsoft:** Highlight, copy from dropdown menu, paste from dropdown menu or Highlight, CTRL+C to copy, CTRL+V to paste

GUI-6$^{\#}$ The application shall permit resizing of application windows. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION
Window resizing can be useful to allow the user to customize the appearance of the desktop or to enlarge a window to display more information. The application design should permit resizing for windows for which resizing may be useful. Conversely, some windows (e.g., pop-up status windows and copyright windows) do not require the capability to resize.
TEST METHOD
The application will be exercised to examine the windows displayed by the application. The capability to resize each window will be observed and the suitability or need for resizing will be evaluated.

GUI-7$^{\#}$ A hyperlink shall not navigate to itself. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 3-4

REQUIREMENT CLARIFICATION
When a link is selected, the action is to load a new page that is either in the same application or in a different application. A link does not navigate to itself (i.e., to the top of the page in which the link appears). The link should not navigate to the same visible portion of a document (i.e., the link is visible on the user's screen); the link can navigate to a different portion of the same document, thus saving the user time to scroll down to that point. Each link on a page navigates to a different destination; the same link is not repeated with different names.

TEST METHOD
Links on the application home pages and on various sub-pages will be selected to verify that the current page is not the destination of the link. The requirement is met if selecting any link does not result in the same viewable portion of a document being visible in the resulting displayed page. If the application does not use a browser this requirements is Not Applicable.

GUI-8 Not applicable for Version 3.0 and above test procedures.

3.6 INTEGRATION SECURITY

INTSEC-1$^{\#}$*The directories touched during the application installation shall not contain files or directories that are world-writeable as a result of installation of the application. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION

The intent of this requirement is to ensure that the installation of an application does not result in the presence of files or directories in the application directory tree that are world-writeable. On the Unix system this can happen inadvertently due to an incorrectly set umask or because of an incorrectly designed installation procedure.

It is also possible that some files or directories in the application's directory tree should be world-writeable. This is acceptable provided such files or directories do not introduce security vulnerabilities. These files and directories should be identified in the application installation and security documentation.

On Microsoft OSs, by default every user belongs to a group called "everyone". The "everyone" group by default has "full" access to some files on the system. For applications using IIS (web server) and additional anonymous internet user is added called IUSR_Server. The neither user should be given Full or Change access to a file or directory by the application unless fully documented as stated above.

TEST METHOD

The following command can be used to scan the application directory tree for world-writeable files:

UNIX:

```
    find root_dir –perm -0002
```

where root_dir is the root of the application directory tree. The –perm option of -0002 will match all files and directories that are world-writeable. This command can be piped to the input of another command as necessary.

Microsoft:

Pre and post installation runs of “cacls” should have been executed at this point and their output placed in “pre_cacls.txt” and “post_cacls.txt” respectively.

The test engineer will compare the files using the command:

```
fc \results\pre_cacls.txt \results\post_cacls.txt >> \results\cacls_diff.txt <-
```

By comparing the files(\temp\pre_cacls.txt with \temp\post_cacls.txt), the test engineer will verify that the application does not allow the ‘Everyone’ group Full(Everyone:F) or Change(Everyone:C) access to files added or touched by the application installation. In the case of applications utilizing IIS (web server) the test engineer will also verify that the application does not allow the ‘IUSR_<servername>’ group (Anonymous Internet User) Full or Change access to files added or touched by the application installation.

This requirement will not be met if there are world-writeable files or directories in the application directory tree that have consequences for either the security of the application or the security of the platform.

This requirement is not met if there are world-writeable files or directories in the application directory tree that have consequences for either the security of the application or the security of the platform.

INTSEC-2$^{\#}$*The application shall not require software development tools on functional user workstations. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION

The presence or absence of software development tools on workstations or servers is a site security policy item.

Development tools include tools that compile source code into executable objects, tools that interpret and execute source code files, and tools that are used to trace and debug an executing object. The intent of this requirement is to prevent users from modifying the intended behavior of an application and from introducing new executable objects onto a workstation.

Compilers and compiler support software (e.g., the C and C++ compilers) are not permitted on general user workstations. The

execution of compiled software objects does not require the presence of these tools. Compilers for mobile code such as Java are included in this group. Likewise, software debuggers are not needed to execute the application. A debugger might be used to modify the behavior of the application and should not be available on user workstations.
Interpreter software, such as perl or TCL/TK, are necessary in order to launch and run software written in those languages. Any mission application software that includes interpreted software must be adequately protected from alteration. Development tools may be required on certain systems, such as development systems. The site security concept of operations must address this issue. However, functional users must not need them in order to use the application.

TEST METHOD

The application configuration and installation guide will be examined to verify that software development tools are not required to use the application. The application will be installed on workstations that are loaded with the standard common infrastructure that does not include software development tools. Following installation of the application, the directories that have been touched by the application installation will be examined to verify that no software development tools have been added to the workstation. Tools that are not permitted on user systems include:

-Compilers (e.g. cc, c++, javac, f77, RATFOR)
-Debuggers (e.g. dbx, adb, sdb)

The results files generated by either JAT or the documented manual procedures will be used to expedite the application examination. Examine the application directory tree and execute the command:

UNIX: find / -ctime -*n* | xargs file | grep "executable SPARC" | cut -f1 -d: | xargs ls –acld (Where *n* is the number of days ago the application installation was started.)

Microsoft:

NT:

Select “Start” → “Find” → “For Files or Folders …”
In the “Name & Location” tab:
In the “Named” text box enter “.exe”
I n the “Look in” pull down selection box select all hard drives on the system”
In the “Date Modified” tab:

Select the “Find all files created or modified:” radio button
Select the “between” radio button
In the “between” text box enter the date the application installation started.
In the “and” text box enter the date the application installation was completed.
Select the “Find Now” button
(Repeat above steps looking for “.com” and “.dll” in place of “.exe”)

2000:

Select “Start” → “Search” → “For Files or Folders …”
In the “Search for files or folders named:” text box enter “.exe”
In the “Look in” pull down selection box select all hard drives on the system”
Select “Search Options >>”
Select “Date” check box
Select “files Created” from the pull down text box
Select the “between” radio button
In the “between” text box enter the date the application installation started.
In the “and” text box enter the date the application installation was completed.
Select the “Search Now” button
(Repeat above steps looking for “.com” and “.dll” in place of “.exe”)

XP:

Select “Start” → “Search”
or
Select “Start” → “Search” → “For Files or Folders …” (Classic Start menu)
In the left hand panel of the “Search Results” window select “All files and folders”
In the “All or part of the file name” text box enter “.exe”
In the “Look in” pull down selection box select all hard drives on the system”
Select “When was it modified?”
Select “Specify dates”

In the pull down selection box Select “Created Date” In the “from” text box enter the date the application installation started. In the “to” text box enter the date the application installation was completed. Select the “Search” button (Repeat above steps looking for “.com” and “.dll” in place of “.exe”) The presence of interpreters for perl, TCL/TK, or other scripting languages is acceptable. However, any mission application script that is interpreted and executed should be examined to ensure that its permissions do not permit unauthorized modification.

INTSEC-3# The application shall not implement or require storage of passwords in clear text. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION
In order to simplify or speed up user access to application server applications, the application may implement storage of passwords for transmission to server applications. However, for obvious security reasons, these passwords must not be stored in clear text. This is particularly critical if general users can read the stored information without acquiring any additional privileges.
TEST METHOD
During installation and configuration of the application, the test engineer will verify that the application stores passwords for general users and identify the storage locations. The test engineer will examine the storage locations and view the passwords. The requirement is not met if the passwords are stored in clear text.

INTSEC-4*The application shall not require the presence of an entry relating to the application server in the /.rhosts file. (UNIX only)

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION
Entries in the /.rhosts file should be made with care since several security vulnerabilities can be traced to incorrect usage of this file.

Depending upon the site security architecture and the application design, an entry in the /.rhosts file may be appropriate. However, using the /.rhosts file is discouraged in most cases; therefore the entries should be kept to a minimum. Using the /.rhosts file to permit transparent access by root from remote workstations should be avoided unless absolutely necessary. Instead, the access should be mapped to another user ID.

TEST METHOD

The /.rhosts file on the test workstation(s) will be examined for entries corresponding to the application server. If such entries are found, they will be removed to determine if application requires the deleted entries to function correctly.

This requirement is Not Applicable for the Microsoft OSs, since there is no equivalent /.rhosts file.

INTSEC-5[#] The application shall use system access control facilities for discretionary access. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-4

REQUIREMENT CLARIFICATION

In general, applications must rely on the security services provided by the common infrastructure instead of duplicating them. An application will only implement security functions that are unique to it if the protection mechanisms of the platform OS are considered inadequate or unacceptable for discretionary access control (DAC). Application PMOs must confirm such requirements and obtain approval from the DoDIIS Engineering Review Board (ERB) and the application security certifier before implementing additional DAC.

In addition, DAC may be implemented in different ways. For example, applications may implement a login process as part of the authentication policy, or control access based on IP address or group membership.

Whatever DAC is implemented, the user must not be permitted to access the IMA without first meeting the implemented access restrictions.

TEST METHOD

If an application implements security functions that are unique to it that cannot be met by the infrastructure security services, appropriate application documentation, e.g., System Security Requirements, System Security Analysis, will be examined to determine the implementation and necessity of the specialized discretionary access.

The application will be exercised to verify that application and system resources are protected by discretionary access controls in accordance with DoDIIS requirements.

Examples of access controls are : login account and passwords protection, restricted object access permissions, access control lists, and privileged user roles.

The test engineer will attempt to bypass the aforementioned controls to gain unauthorized access to data or processes, some examples for doing so are given below.

If a web server implements login as part of the application identification and authentication policy, prior to logging in to the application, the test engineer will exercise hyperlinks and enter absolute paths in the destination field of the browser.

The requirement is met if each attempt to use the hyperlink or absolute path is either denied or the test engineer is presented the application login page.

If DAC is implemented by restrictions to specific IP addresses, hostnames, or groups of addresses and hostnames then the test engineer will attempt to access these hyperlinks and absolute paths from inappropriate IP addresses or hostnames.

The requirement is met if each attempt to use the hyperlink or absolute path is denied.

To determine if access controls are implemented on Apache web servers:

Check for entries in Apache's access.conf and httpd.conf files in the "conf" directory of the Apache application's home directory tree.

Additionally, any Apache .htaccess files can define access control per directory and can modify the global directives contained in access.conf or httpd.conf. If present, .htaccess files will be in the directories under the document root of the server document directory tree. The .htaccess files act to restrict access to that directory unless the rules specified by directives within the access.conf and httpd.conf files are met.

For Netscape servers, examine the “authenticate” directives within the *<server_root>*/*<server_ID*/conf/magnus.conf file. Microsoft IIS uses a WebDAV (Distributed Authoring and Versioning) mechanism to manage DAC services. If WebDAV is implemented, navigate to the WebDAV directory, usually under the Intepub directory, and view the permissions on the directory. By default, the assigned permissions are “Full Control” for everyone. Confirm that permissions are less open than the default. Based upon the application design and implementation, ad hoc test cases will be run by the test team to exercise and demonstrate the discretionary access functions of the application.

INTSEC-6# The application shall not require users to login using privileged user accounts. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION
General users must not need to login as root or as a privileged user (e.g., an administrative user on Microsoft OSs) to perform general user functions. While specific application functions may require execution with additional privileges, the privilege can be granted on demand by the application in a way that is transparent to the user. Additional privileges may be required to manage the application. Users who perform management of the system’s resources or who are responsible for the security of the system are the only individuals who should have access to root privileges or to other system privileges.
TEST METHOD
The appropriate application documentation (e.g., SDD, Software User’s Manual (SUM)) will be examined to verify that login as root or as a privileged user is not required to use the application. The test engineer will login to the application as a general user, following the configuration and installation of the application. The test engineer will perform ad hoc tests to verify the basic function of the application.

INTSEC-7# The application shall not require functional user access to a shell or command prompt. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-3

REQUIREMENT CLARIFICATION

Although restriction of shell or command prompt access is no longer considered a security requirement, uncontrolled use of the shell or command prompt should be discouraged. This not only prevents users from taking advantage of vulnerabilities of the OS or workstation configuration, but also reduces the possibility of users damaging either data or environment by incorrect usage of Unix/ Microsoft OSs capabilities. Instead, user interaction with the application should be through graphical user interfaces.

TEST METHOD

The appropriate application documentation (e.g., SUM) will be examined to identify how a user invokes and executes the application. The documentation will verify that shell or command prompt access is not required to use the application. Following configuration and installation of the application, invoke the application. Execute ad hoc test cases to verify that the application will execute properly without the use of a shell or command prompt.

INTSEC-8*Application programs shall not be assigned setuid or setgid permissions to another user ID or group ID. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION

Programs that are configured to execute with setuid/setgid/Log on as/This Account, etc., are a source of potential security vulnerabilities in site workstations and servers, particularly if the application is configured to execute with privileged access, e.g.: ‘root’ or ‘administrator’. The need to configure applications thusly should be stated clearly in the application design documentation.

TEST METHOD

Following the configuration and installation of the application, the permissions that are set on the application executable files/services will be reviewed to verify that they are not set to assume altered user/group IDs upon execution. Additionally, executables configured to run as altered users/groups that have restricted permissions will not necessarily fail this requirement.

Unix:

Locate suid and sgid files by issuing the following commands:

```
# cd <APPLICATION_ROOT>
```

```
#find . –perm -4000 –ls          ;returns set UID files
#find . –perm -2000 –ls          ;returns set GID files
```

For each file that has setuid and/or setgid, the exact permissions will be noted.

Microsoft:

The data for every instance of HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\[service name]\ObjectName will be reviewed to ensure the service is not configured to run with privileges that have potential to cause a security vulnerability. Common acceptable values are, but not limited to, 'LocalSystem', 'Network Authority\NetworkService' and 'Network Authority\LocalService'.

INTSEC-9[#] Operation of the application shall not modify OS and other shared files. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION

In general, execution of the application should not create security vulnerabilities for other applications or for the OS of the user's workstation or of the platform on which the application server resides. Vulnerabilities could occur due to changes in permissions of application files, changes in ownership of application files or other files, or modification of the contents of application files and files shared with other applications. This requirement applies to all phases of application usage, i.e., startup and initialization, information processing, logging/auditing, and application termination. This also includes the capability of obtaining a command line prompt (e.g., a UNIX shell) from within the application. While access to the command line may not be prohibited, it is a service of the infrastructure, not of the application, and such a capability might allow a user to modify resources without authorization.

TEST METHOD

The application documentation will be reviewed to determine the application files and other shared files that are referenced by the application during normal use.
Output of the truss command, (e.g. truss –f –e –a –o *output file* [*application_name OR –p process_id])* should be examined for modification of shared files, as well. The requirement is not met if a file written by the application contains system-wide resources that would create security vulnerabilities for other applications or for the OS of the user's workstation.

INTSEC-10 The application shall not implement audit collection or audit delivery functions. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-3

REQUIREMENT CLARIFICATION
The common infrastructure provides an audit API for applications. Applications that use this API do not have any need to implement additional audit functionality.
TEST METHOD
The appropriate application documentation (e.g., System Security Requirements, System Security Analysis) will be examined to determine the use of the infrastructure audit API for generating audit records. The application will be inspected to verify that audit collection or audit delivery functions are not implemented by the application.

INTSEC-11 The application shall use the infrastructure audit API for generating audit records. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 2-3

REQUIREMENT CLARIFICATION
The common infrastructure provides a set of security functions. This set includes a single audit API for use by applications to write and transmit audit records. Therefore, there is no need for an application to either use a different audit mechanism or to implement its own unique audit mechanism.
TEST METHOD
The appropriate application documentation (e.g., System Security Requirements/Analysis) will be examined to determine that audit API is being used for generating audit records by the application. UNIX: To verify the use of the audit API for generating audit records by the application execute the following command in a shell window: *tail –f /var/log/syslog* (The lines are displayed in the window as applications; application utilities write them to the syslog file. Using selected test cases from the application security test procedures, verify that application audits are written to **/var/log/syslog** and displayed in the shell window.)

The audit API generates audit records in the following format:
DTG: (month, day, and time the audit record was generated)
Process Name [PID]: (the name of the workstation or server on which the process is running)
Program: (ASCII name of the process that generates the message)
Program Event ID: (the numeric ID associated with the audit event)
Message Level: (an ASCII keyword that indicates the urgency level of the audit record)
User Name [UID]: (the ASCII name and numeric user ID of the general user that owns the process generating the message)
Event Specific Information\n (determined by the security requirements of the application and must be terminated with the new line character, '\n')

(Ex. Mar 10 14:33:41 machine1 in.rlogind[4376]: CSE:23:INFO:root[0]:connect from machine2)

Microsoft**:**
The Event Log is used to store audit information from an application.

NT: From the Start menu select: Programs->Administrative Tools->Event Viewer
Select Application from the list. (Application logs are displayed.)

All application logs are displayed.2000 and XP: From the Start menu select:
Settings->Control Panel->Administrative Tools->Event Viewer. Select Application from the list. (Application logs are displayed.)

This requirement is not met if the application writes no audits.

INTSEC-12 The application audit strategy shall be integrated into site audit architecture. (UNIX only)

IMPACT CODE RANGE: 2-3

REQUIREMENT CLARIFICATION
The use of the infrastructure audit interface is required. Compliance with this requirement is an important step toward integrating the

application auditing into the site audit architecture. This is because all applications that comply with this requirement will be using the same audit (API) and the same audit formats. This uniformity will improve the ability of a site to implement a single approach to audit collection and analysis.

A site’s audit strategy will also include collection and analysis of OS audit data. An application may either rely on the OS auditing or actually generate audits that use the OS audit API. The approach should be clearly documented in the application design documentation, and the audit collection mechanism, API, and audit formats should be clearly described.

TEST METHOD

The primary consideration in evaluating if an application meets this requirement is the level of effort required to integrate the application’s audit into a site’s audit architecture. A strategy that does not use either the infrastructure audit API or the OS audit API does not meet this requirement. Reliance on the OS API can pose difficulties since the audit API and audit format will differ across the OSs. Since the OS audits must also be integrated into the site audit architecture, this approach is acceptable. However, it poses a level of effort that is higher than the use of the infrastructure audit API.

For Microsoft OSs, auditing is done automatically by the OS. Therefore, this requirement is Not Applicable.

INTSEC-13 Not applicable for Version 4.0 and above test procedures. Incorporated into INTSEC-16.

INTSEC-14# The application web server shall not store sensitive information in cookies. (UNIX and Microsoft OSs)

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION

Although security policy does not prevent the use of cookies, an application should not write sensitive information to the cookie file. Sensitive information is any information, such as the user’s password, that may affect the security posture of the application or of other site systems.

TEST METHOD

The application will be exercised from a client workstation. The browser in use on the workstation will be configured to accept cookies. During the user’s session with the application server, the browser cookie file will be monitored and the contents of each cookie written by the browser will be examined for potential vulnerabilities.

File Format:

Column 1	Domain or host name of server sending the cookie
Column 2	Code for whether first column value represents host name or domain name (domain=TRUE)
Column 3	Virtual or partial path for host name or domain name specified.
Column 4	Is a secured socket connection (SSL) required? (yes=TRUE)
Column 5	Time of expiration
Column 6	Name of cookie
Column 7	Value of cookie

If the application does not use a web server this requirement is Not Applicable.

INTSEC-15# - Not applicable for Version 4.2 and above test procedures. Incorporated into INTSEC-5

INTSEC-16#*The web server shall log all connections and data requests that are received by the web server including auditing user activity in accordance with DoDIIS security policy. (UNIX and Microsoft OSs). NOTE: This requirement was identified as INST-30 in Version 2.1.

IMPACT CODE RANGE: 2

REQUIREMENT CLARIFICATION
Logging by the server, including auditing user activity, assists in identifying operational problems as well as providing a record of access to the server The audit log record should include the date and time, the host name, the files or services accessed, and, if possible, the username.

TEST METHOD

The application server configuration files will be examined to verify that logging by the http daemon is properly configured, including auditing user activity in accordance with DoDIIS security policy.

Application documentation will be reviewed to identify the logging and auditing strategy of the application web server.
The application will be exercised from a client workstation. The audit and log trails of the application server will be monitored to verify that the application server is logging all connections and data requests as well as auditing user activity.

Web servers should use the Common Logfile Format (CLF) for audit and access logs. The default format suffices for most purposes, and should be used whenever possible to ensure compatibility with common log parsing software. Apache and Netscape servers write to CLF by default; Microsoft IIS servers can do the same by selecting a configuration option within the Graphical User Interface (GUI). The file format is described below:

Format:
remote host local host authuser date request status bytes

Explanation:

remote host	IP address of workstation or server requesting access
local host	IP address of local web server (normally blank)
authuser	ID for authenticated user. Will be blank if no login is required
date	Date and time of request, enclosed within brackets
request	HTTP request. Contains method (usually GET) and page title
status	HTTP status code. Codes are defined in HTTP specification.
Bytes	Bytes returned. Same as file size of page requested.

Example:

```
192.9.200.1 - - [8 May/2001:06:38:00 -0600] “GET /index.html HTTP/1.0” 200 5248
```

Items are separated by a single space; data items containing spaces are encapsulated within either brackets or quotes, as seen above. Blank fields will show a dash (-) as a placeholder, to assist the log parsers in correctly displaying log data. For servers not using authentication, the second and third fields will normally be blank. Servers using authentication services will require use of the ident daemon on Unix systems, and will populate the second and third fields.

UNIX – Apache

```
# cd [web server base directory]/conf
# grep ^CustomLog *conf      ;note the log file.
   (e.g., interpreting the following result from the ‘grep’ command:
         httpd.conf:CustomLog /opt/apache/logs/access_log  common
         the log file is ‘/opt/apache/logs/access_log’)
# view [log file]
-verify that the required data is being logged into the log file.
```

For centralized servers with browser access this requirement is evaluated on the server.

If the application does not use a web server, this requirement is Not Applicable.

INTSEC-17 Not applicable for Version 4.0 and above test procedures. Incorporated into INTSEC-15.

INTSEC-18$^{\#}$*****The web server processes shall be owned and run by a user name that is not superuser (UNIX) or an administrative user (Microsoft OSs). (UNIX and Microsoft OSs). NOTE: This requirement was identified as INST-32 in Version 2.1.

IMPACT CODE RANGE: 1-2

REQUIREMENT CLARIFICATION

Files, directories, and processes that are not directly related to OS and platform management should not be owned by a superuser (root on UNIX and an administrator user on Microsoft OSs) to limit security vulnerabilities and to avoid the need for superuser access to manage the application.

TEST METHOD

The ownership of the httpd executable file shall be examined to verify that it is not owned by root (UNIX) or an administrative user (Microsoft OSs).

After the http daemon has started, the ownership of the httpd process shall be inspected to verify that it is not owned by root (UNIX) or an administrative user (Microsoft OSs).

(UNIX– Apache)

There are 3 configuration files, (httpd.conf, srm.conf and access.conf), that can contain these server settings. The following commands will return the appropriate settings that should be compared:

```
# cd <HTTP server root directory>/conf/
# grep “^User  “ *.conf   (note the single space
    between the ‘r’ and quote)
```

(Netscape Servers)

Verify that the ownership of the httpd, ns-httpd and uxwdog processes are not owned by root.

For centralized servers with browser access this requirement is evaluated on the server.

This requirement is Not Applicable if the application does not use a web server.

INTSEC- 19# General users shall not view or launch privileged application functions. (UNIX and Microsoft OSs).

IMPACT CODE RANGE: 1-3

REQUIREMENT CLARIFICATION
In keeping with the security principle of least privilege, a general user should only be presented with selections or functions that he/she is authorized to access. Privileged functions should not appear on a user's menu if they cannot be selected. This approach reduces the possibility of unauthorized users exploiting application functions that can affect the security of the application or infrastructure.
TEST METHOD
Tester will access the application as a general user. The menus and function selections will be evaluated to verify that a general user cannot view privileged functions.

INTSEC-20 Upon successful completion and verification of the application installation, a security risk analysis will be conducted to target system vulnerabilities. (UNIX and Microsoft OSs).

IMPACT CODE RANGE: 1-4

REQUIREMENT CLARIFICATION
A security risk analysis will be conducted to identify vulnerabilities. This activity will assess the security of systems to be fielded, pre-deployment, with greater control and accountability, so that issues can be efficiently identified and resolved in conjunction with BETA-I test.
TEST METHOD
The JITF Vulnerability Assessment Team, as a component of DIA SYS-4 Vulnerability Assessment Team, will use analysis tools to gather data and generate reports assessing risk. Information gathered during this assessment will be included as an addendum to the JITF test report and will be included in the security certifier's assessment, as well.

4. OPERATING SYSTEM PATCH AND ADVISORIES ASSESSMENTS

The JITF receives alerts and advisories regarding OSs and other software from many sources. The JITF tracks these bulletins and reviews weekly the patches and advisories for the Solaris and Microsoft OSs and other software used in the common infrastructures. Those of possible impact and relevance to CSE and AFDI systems are examined in depth and installed on test servers or workstations. The JITF evaluates the effects of the patches on the infrastructure and publishes reports via the VTF.

The reports will contain, when possible, the nature of the vulnerability, type of exploit, and solution to the problem, as well as any impact to the CSE or AFDI infrastructures. The JITF will work with CSE and AFDI developers to resolve any problems created by the patch under examination and will also coordinate with DIA/SY-S4 to resolve any conflicts between integration and information assurance requirements.

5. ACRONYMS

ACRONYM	DEFINITION
ABI	Application Binary Interface
AC2ISRC	Aerospace Command and Control Intelligence Surveillance Reconnaissance Center
AFDI	Air Force DoDIIS Infrastructure
AFRL	Air Force Research Laboratory
API	Application Program Interface
CLF	Common Logfile Format
CM	Configuration Management
COTS	Commercial Off-The-Shelf
CUBIC	Common User Baseline for the Intelligence Community
DAC	Discretionary Access Control
DBMS	Data Base Management System
DexA	DODIIS Executive Agent
DII COE	Defense Information Infrastructure Common Operating Environment
DMB	DoDIIS Management Board
DoDIIS	Department of Defense Intelligence Information System
ERB	Engineering Review Board
FAT	Factory Acceptance Test
GIF	Graphics Interchange Format
GOTS	Government Off-The-Shelf
GUI	Graphical User Interface
html	Hyper Text Markup Language
http	Hyper Text Transfer Protocol
ID	Identifier
IMA	Intelligence Mission Application

ACRONYM	DEFINITION
IMS	Information Management Services
IP	Internet Protocol
ITF	Integration Test Facility
JAT	JITF Automated Tool
JITF	Joint Integration Test Facility
JTA	Joint Technical Architecture
JTPM	Joint Test Planning Meeting
NFS	Network File System
NIMA	National Imagery and Mapping Agency
NIS	Network Information Service
OS	Operating System
PID	Process Identifier
POC	Point Of Contact
PMO	Program Management Office
RAM	Random Access Memory
RPC	Remote Procedure Call
SDD	Software Design Document
SUM	Software User’s Manual
TCP	Transmission Control Protocol
TFUG	Trusted Facility User’s Guide
URL	Uniform Resource Locator
VDD	Version Description Document
VTF	Virtual Test Folder
WebDAV	Web Distributed Authoring and Versioning
XPG	X/OPEN Portability Guide

6. DEFINITION OF TERMS

Application Administrator - A user who has access to privileged functions associated with the maintenance and management of an individual application and its users.

Application Baseline - A fixed set of files necessary to operate an application.

Application Server - A workstation that has been designated to provide the files and processes necessary to execute an application.

Common Infrastructure - A set of basic data and services provided as a shared resource to applications for the purpose of minimizing redundancy and facilitating integration and interoperability of applications.

Common Operating Environment - a common information technology architecture that promotes interoperability and cross-platform capabilities.

General User- A user who does not have access to privileged functions.

Information Technology Components - Software or portions of software that may be introduced into an information systems environment.

Infrastructure Application Selection Mechanism - An icon or menu item provided by the existing infrastructure environment that initiates the launch of a software application.

Infrastructure Compliance - The ability of a software application to operate within the guidelines provided by integration, interoperability, and security requirements.

Installation and Configuration Guide - A set of instructions that include steps to successfully load a software application and customize its use.

Integrating Quality - The extent to which an application is able to be introduced and cohabit in an existing system environment.

Intelligence Mission Application (IMA) – An IMA is a software module or set of software modules that implement an intelligence mission function. IMA architecture can be based on one of several configurations including: client/server and web based applications with either thick or thin clients.

Multi-tiered Operating Environment - an information technology system that is composed of several layers - e.g., a presentation layer (the browser), business rules (the server), and storage (the database).

Site Administrator - A privileged user responsible for coordination, management, and maintenance of all information resources at a particular geographic location.

Trusted User - A user who has been granted a privileged role that may include access to system control, monitoring, or administration functions.

7. PRE AND POST EVALUATION PROCEDURES

If JAT is not available the test engineer will perform the following steps to create "snapshots" that will capture system changes. In either case a results directory will be created. This directory can be used to expedite searching and will be referred to in the test methods of applicable requirements found in Section 3.1.

	Before load of the IMA Close all applications/windows and assume root or administrative Privileges.	**After load of the IMA** Close all applications/windows and assume root or administrative Privileges.
Unix	# /usr/bin/cd / <-- # /usr/bin/mkdir results <-- # /usr/bin/mv /results/pre_list.txt /results/pre_list_txt.orig <-- (if they exist) # /usr/bin/find / -follow ! -local -prune -o -name net -prune -o -name proc -prune -o -ls >> /results/pre_list.txt <--	# /usr/bin/cd / <-- # /usr/bin/mv /results/post_list.txt /results/post_list_txt.orig <-- (if they exist) # /usr/bin/find / -follow ! -local -prune -o -name net -prune -o -name proc -prune -o -ls >> /results/post_list.txt <-- Compare the files (/results/pre_list.txt with /results/post_list.txt) and review the differences against the requirements.
Microsoft	Start -> Run. In the open field enter: Cmd <-- At the command prompt enter: > move \results\pre_cacls.txt \results\pre_cacls_txt.orig <-- (if it exists) > move \results\pre_list.txt \results\pre_list_txt.orig <-- (if it exists)	Start --> Run. In the open field enter: Cmd <- In the command prompt enter: > move \results\post_cacls.txt \results\post_cacls_txt.orig <- (if it exists) > move \results\post_list.txt \results\post_list_txt.orig <- (if it exists) > move \results\post_reg.txt \results\post_reg_txt.orig <- (if it

	> move \results\pre_reg.txt \results\pre_reg_txt.orig <-- (if it exists) Perform the following steps for each logical disk drive where <drive> is the letter of the drive: > mkdir \results > cd <drive>:\ > (FOR /R %j IN (*) DO CACLS "%j" /c) >> \results\pre_cacls.txt <-- > dir /s /t:w /a <drive>: 2>>\results\pre_list.err >>\results\pre_list.txt <-- Then, at the command prompt record the registry > Regedit /e \results\pre_reg.txt <-- (Windows NT) or > Regedit /a \results\pre_reg.txt <-- (Windows 2000/XP)	exists) Perform the following steps for each logical disk drive where <drive> is the letter of the drive: > (FOR /R drive: %j IN (*) DO CACLS "%j" /c) >> \results\post_cacls.txt <- > dir /s /t:w /a drive: 2>>\results\post_list.err >>\results\post_list.txt <- Then, at the command prompt record the registry > Regedit /e \results\post_reg.txt <- (Windows NT) > Regedit /a \results\post_reg.txt <- (Windows 2000/XP) Compare the files (\results\pre_cacls.txt with \results\post_cacls.txt, \results\pre_list.txt with \results\post_list.txt, and \results\pre_reg.txt with \results\post_reg.txt) > del \results*diff.txt <- (if they exist) > fc \results\pre_cacls.txt \results\post_cacls.txt >> \results\cacls_diff.txt <- > fc \results\pre_list.txt \results\post_list.txt >> \results\file_diff.txt <- > fc \results\pre_reg.txt \results\post_reg.txt >> \results\reg_diff.txt <- Review differences against requirements.